Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

1972-03

A GPSS-like simulator callable from a FORTRAN program.

Williams, Robert Joseph

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/40229

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A GPSS-LIKE SIMULATOR

CALLABLE FROM A FORTRAN PROGRAM

Robert Joseph Williams

Thesis Advisor: G. E. Heidorn

March 1972

Thesis
W6215

Approved for public release; distribution unlimited.

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A GPSS-LIKE SIMULATOR

CALLABLE FROM A FORTRAN PROGRAM

Robert Joseph Williams

Thesis Advisor: G. E. Heidorn

March 1972

Approved for public release; distribution unlimited.

T146654

A GPSS-like Simulator

Callable from a FORTRAN Program

by

Robert Joseph Williams
Lieutenant Commander, Supply Corps, United States Navy
B.S., University of Scranton, 1961

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1972

Author [redacted]

Approved by: [redacted]
Thesis Advisor

[redacted]
Chairman, Department of Operations Research
and Administrative Sciences

[redacted]
Academic Dean

ABSTRACT

As part of an on-line simulation analysis system being developed at the Naval Postgraduate School there was a need for a GPSS-like simulator which could be run under control of a FORTRAN program. The objective of the work done for this thesis was to design an information structure appropriate for such a simulator and to write the simulation routine. This thesis gives a detailed description of the information structure, telling what it must look like when the routine is called and what it contains when the routine is through. Another section discusses the procedure by which the routine performs the simulation. Program listings and a complete sample problem are included, also.

TABLE OF CONTENTS

I. INTRODUCTION ---------- 7

II. THE INFORMATION STRUCTURE ---------- 9

A. FIRST 32 ELEMENTS OF THE X-VECTOR ---------- 10

B. STORAGES ---------- 15

C. QUEUES ---------- 17

D. BLOCKS ---------- 18

E. TABLES ---------- 21

F. FUNCTIONS ---------- 23

G. VARIABLES ---------- 26

H. TRANSACTIONS ---------- 27

I. A SAMPLE PROBLEM ---------- 28

III. THE PROCEDURE ---------- 30

A. OVERALL FLOW OF ROUTINE ---------- 30

B. SCAN AND MOVE TRANSACTION ---------- 31

C. EVALUATE ARGUMENTS ---------- 32

1. Variable Evaluation ---------- 34

2. Function Evaluation ---------- 35

D. BLOCK ROUTINES ---------- 36

E. UPDATE CLOCK ---------- 37

F. TRACING VARIABLES ---------- 37

G. ERROR CODES ---------- 38

IV. CONCLUSIONS AND RECOMMENDATIONS ---------- 39

APPENDIX A (SAMPLE PROBLEM) ---------- 57

APPENDIX B (PROGRAM VARIABLE DICTIONARY)---------- 72

SIMULATION ROUTINE FORTRAN LISTING ---------- 76

LIST OF REFERENCES ---------- 104

INITIAL DISTRIBUTION LIST ---------- 105

FORM DD 1473 ---------- 106

LIST OF FIGURES

1. X-VECTOR GENERAL LAYOUT ---------- 40
2. ERROR CODES ---------- 41
3. STORAGE ENTITY LAYOUT ---------- 42
4. QUEUE ENTITY LAYOUT ---------- 43
5. BLOCK ENTITY LAYOUT ---------- 44
6. BLOCK TYPE CODES AND BLOCK ARGUMENTS ---------- 45
7. STANDARD NUMERICAL ATTRIBUTE (SNA) CODES ---------- 47
8. MODE CODES ---------- 49
9. TABLE ENTITY LAYOUT ---------- 50
10. FUNCTION ENTITY LAYOUT ---------- 51
11. TRANSACTION ENTITY LAYOUT ---------- 52
12. FLOWCHART OF SCAN ROUTINE ---------- 53
13. FLOWCHART OF MOVE TRANSACTION ROUTINE ---------- 54
14. FLOWCHART OF UPDATE CLOCK ROUTINE ---------- 55
15. FLOWCHART OF ARGUMENT EVALUATION ROUTINE ---------- 56

TABLE OF ABBREVIATIONS

Abbreviation	Meaning
BDT	Block Departure Time
CEC	Current Event Chain
CURR	Current
DEC	Decimal
FEC	Future Event Chain
FTCEC	First Transaction on Current Events Chain
FTFEC	First Transaction on Future Events Chain
GPSS	General Purpose Simulation System
INT	Integer
INTDEC	Integer/Decimal
IPD	Internal Problem Definition
LTCEC	Last Transaction Current Events Chain
LTFEC	Last Transaction Future Events Chain
OPCODE	Block Type Code
OS/FOS	Operand Stack/Floating Point Operand Stack
RAS	Return Address Stack
SCFLAG	Status Change Flag
SSIND	Scan Status Indicator
SNA	Standard Numerical Attribute

I. INTRODUCTION

A research project at the Naval Postgraduate School has as its goal the development of a system with which simulation analyses can be performed through natural language dialogue with a computer [1]. Currently, the capability exists for translating an English description of a queueing problem into an Internal Problem Description (IPD) and for translating an IPD into an equivalent English description and into a GPSS simulation program [1, 2, 3]. The simulation can then be performed by running the GPSS program [4, 5].

It would be desirable in the system being developed to provide for analyst-computer interaction during the simulation, but the current mode of operation does not allow this, because of the GPSS program having to be run separately. In order to make this interactive capability possible, it was decided to implement a GPSS-like simulator that could be run under control of the current FORTRAN program.

A. OBJECTIVE

The objective of the work done for this thesis, then, was to design an information structure appropriate for the simulator and to write a simulation routine. A design goal was to make the information structure as compact and self-contained as possible. This was accomplished by using one singly-subscripted variable common to

the simulation routine and the calling program to hold all of the information about the model.

The information in this array is structured in basically the same manner as the "internal tables" of the GPSS simulator, and the algorithm of the simulation routine is basically the same as that of GPSS [5]. Although this routine is intended for use with the system described earlier, it could be called from any FORTRAN program.

B. ORDER OF REPORTING

Section II of this thesis describes the information structure in detail. It tells what must be there when the routine is called and what can be found there when it is through. Section III discusses the procedure by which the routine performs the simulation. Section IV gives conclusions and recommendations. A complete sample problem is contained in Appendix A.

II. THE INFORMATION STRUCTURE

In the Simulation Routine all information about a model is held in a one-dimensional array called the X-Vector, each element of which is a four-byte word in the IBM 360. The allocation of elements in this vector is completely flexible, and, except for the first 32 elements, depends entirely on the particular model being specified. For example, if only one Storage is used in the model, then only enough elements for one Storage entity need be allocated, and these elements can be located almost anywhere in the array. The first 11 elements of the X-Vector are used for certain attributes that are needed for all models, and the next 21 elements furnish information about what is in the remainder of the array. Currently, this array is DIMENSIONed to have 500 elements, but this can be altered by changing 2 numbers in the FORTRAN program.

With the exception of Transactions, the X-Vector elements for all entities must be assigned and initialized prior to the calling of the Simulation Routine. Each type of entity (except Transaction and Savevalue) has 2 locations in the first 32 elements of the X-Vector to specify the quantity of that type of entity in the model and to furnish a pointer to the directory for that type of entity. The directory, then, contains a pointer to each entity of that type. Each of these pointers

actually points to the element immediately preceding that entity. The locations of the directories and the entities are arbitrary, except that they must be after element 32 and before the Transactions.

Savevalues do not require a directory, since each one uses only one X-Vector element, and these elements must be contiguous. For transactions all that need be specified is a pointer (X25) to the beginning of the area in the X-Vector where the Transactions are to be. The Simulation Routine initially allocates elements for each Transaction and chains these entities to form a list of unused Transactions.

In this section of the thesis the X-Vector layout for each type of entity is described in detail. Also, the information required in the first 32 elements is specified. It is stated what must be in the X-Vector prior to calling the Simulation Routine and what will be found there after the routine is executed. The sample problem in Appendix A furnishes examples of most of the points discussed in this section.

A. FIRST 32 ELEMENTS OF THE X-VECTOR

These elements contain attributes and pointers which play a major role in controlling the simulation. The following subsections describe the manner in which these elements must be initialized by the calling program. Where a value is changed during the course of the simulation, it is so indicated. If an entity is not defined in a model, then that entity's associated element need not be initialized to any value, since the Simulation Routine will never reference it.

1. Mode (X1)

This element is initialized to the values 1, 2, 3, or 4, according to the following table.

MODE	GPSS
1	Simulate/Start
2	Reset/Start
3	Clear/Start
4	Start

This element performs the same function as the Start, Reset/Start and Clear/Start cards in GPSS. If the Mode is initialized to 1, the Simulation Routine computes slopes, sets the clocks to zero, creates a chain of unused Transactions, and places one Transaction on the FEC for each GENERATE block in the model. If the Mode is initialized to 2 or 3, then the model entities are reset or cleared the same way as by the Reset card or Clear card in GPSS. The simulation then continues from where it left off when it returned control to the calling program. If the Mode is initialized to 4, then the calling program must set the Termination Count to some new value, and return control to the Simulation Routine where the simulation will continue from where it left off.

2. Termination Count (X2)

The Termination Count serves the same purpose as the first argument of a GPSS Start card. It must be initialized to some integer value each time the Simulation Routine is called. Then during the

simulation, whenever, a Transaction enters a TERMINATE block, this value is decremented by the value of argument A of the TERMINATE block. When the Termination Count reaches zero, the simulation stops and control is returned to the calling program.

3. Random Number Seeds (X3-X10)

These elements are initialized to the same values used on a GPSS RMULT card. The values will change during the course of the simulation, since the random number generator stores the current random number for each sequence in these elements.

4. Clock Time (X11)

This element corresponds to the GPSS relative clock.

5. Number of Storages (X12)

This element is initialized to the number of Storage entities used in the model.

6. Storage Directory Pointer (X13)

This element is initialized to the subscript of the element immediately preceding the Storage directory.

7. Number of Queues (X14)

This element is initialized to the number of Queue entities used in the model.

8. Queue Directory Pointer (X15)

This element is initialized to the subscript of the element immediately preceding the Queue directory.

9. Number of Functions (X16)

This element is initialized to the number of Function entities defined in the model.

10. Function Directory Pointer (X17)

This element is initialized to the subscript of the element immediately preceding the Function directory.

11. Number of Blocks (X18)

This element is initialized to the number of Block entities defined in the model.

12. Block Directory Pointer (X19)

This element is initialized to the subscript of the element immediately preceding the Block directory.

13. Number of Tables (X20)

This element is initialized to the number of Table entities defined in the model.

14. Table Directory Pointer (X21)

This element is initialized to the subscript of the element immediately preceding the Table directory.

15. Number of Savevalues (X22)

This element is initialized to the number of Savevalue entities used by the model.

16. Savevalue Pointer (X23)

This element is initialized to the subscript of the element immediately preceding those to be used for Savevalues.

17. Number of Parameters (X24)

This value is initialized to the number of parameters each Transaction in the model must have.

18. Transaction Pointer (X25)

This value is initialized to the subscript of the element immediately preceding the Transaction entities. After all other entities have been assigned X-Vector locations, this element would normally be given the highest subscript assigned to those entities. The program will then assign all the remaining X-Vector elements to the Transaction entities.

19. Pointer to First Transaction on Current Events Chain (X26)

This element is initially set equal to zero. Thereafter, it will contain the pointer to the first Transaction on the Current Events Chain (CEC), or zero is there is no Transaction on the CEC.

20. Pointer to Last Transaction on Current Events Chain (X27)

This value is initially set equal to zero. Thereafter, it contains a pointer to the last Transaction on the CEC, or zero if there is none.

21. Pointer to First Transaction on Future Events Chain (X28)

This value is initially set equal to zero. Thereafter, it contains a pointer to the first Transaction on the Future Events Chain (FEC), or zero if there is none.

22. Pointer to Last Transaction on Future Events Chain (X29)

This value is initially set equal to zero. Thereafter, it will contain a pointer to the last Transaction on the FEC, or zero if there is none.

23. Error Code (X30)

This element is initially set equal to zero. If during the simulation, an error is detected by the routine, an error code will be placed in this element and control will be returned to the calling program. These codes correspond to the error numbers in GPSS, and are listed in Figure 2.

24. Number of Variables (X31)

This element is initialized to the number of Variable entities defined in the model.

25. Variable Directory Pointer (X32)

This element is initialized to the subscript of the element immediately preceding the Variable directory.

B. STORAGES

Each Storage entity defined in the model must be assigned 10 contiguous X-Vector elements as illustrated in Figure 3. These values have the same meaning as in GPSS. The calling program must initialize the second element for each Storage entity to the Storage's capacity and all other elements, except the IPD pointer,[1] to zero.

---

[1] Currently, IPD pointers are completely ignored by the Simulation Routine and can simply be initialized to zero.

The Simulation Routine will make entries in the elements of a Storage each time a Transaction passes through an ENTER or LEAVE block that refers to that Storage. If the amount of Storage units required by a Transaction is less than or equal to the amount available, then the current contents is incremented by that amount and the number of units available is decremented by that amount.

The cumulative time integral is an 8-byte floating point sum of the number of clock units that the Storage was occupied, weighted by the number of units in the Storage during each interval that occupancy was constant. The integral is updated by the Simulation Routine each time the Storage contents changes. The cumulative time integral can be used to compute Storage utilization, which is the cumulative time integral divided by the product of the clock time and the Storage capacity.

The last clock time the Storage changed status contains the clock time at which the time integral was last updated. The clock time is placed in this element each time the Storage changes status, and when the Simulation Routine is called using mode 2 (Reset) or mode 3 (Clear).

Entry count is incremented every time a Transaction enters an ENTER block. The magnitude of the increment is specified by argument B of the ENTER block.

Maximum contents is set equal to current contents each time a Transaction enters an ENTER block and the new current contents is greater than the previous maximum contents.

The remaining 3 elements contain 5 pushdown delay chain pointers and the IPD pointer. These delay chain pointers are associated with the following Storage conditions, (1) full, (2) not empty, (3) empty, (4) not full and (5) reduction in contents. Each time a Transaction enters an ENTER block or a GATE block and fails to advance because the Storage condition associated with that block is not met, the Transaction is placed in the appropriate delay chain and the delay chain pointer associated with that condition is set to point to the Transaction.

C. QUEUES

Each Queue entity defined in the model must be assigned 8 contiguous X-Vector elements as illustrated in Figure 4. These values have the same meaning as in GPSS. The calling program must initialize all Queue elements, except the IPD pointer, to zero. All future entries in these elements will be made by the Simulation Routine.

The last clock time the Queue changed status contains the last clock time at which the cumulative time integral was computed. The clock time is placed in this element each time the Queue changes status and when the Simulation Routine is called in mode 2 (Reset) or mode 3 (Clear).

Total entries is incremented each time a Transaction enters a QUEUE block which refers to the Queue. The magnitude of the increment is specified by argument B of the QUEUE block.

Number of zero-delay entries is incremented whenever a transaction enters and leaves the Queue with no time delay. The magnitude of the increment is given by the value of argument B of the QUEUE block.

The other elements are similar to those of the Storage entities.

D. BLOCKS

Block entities defined in a model describe the various actions which are to take place during the simulation. Each Block may have associated with it from 0 to 7 arguments. In general, these Blocks have the same meaning as Blocks in GPSS, and the arguments perform the same function also.

Figure 5 is a general layout of the elements of a Block entity. From 2 to 9 contiguous X-Vector elements are required for each block, depending on the number of arguments specified. The calling program must initialize these elements prior to calling the Simulation Routine. Throughout the simulation, these values will remain unchanged. Figure 6 is a list of Block type numbers, Block names and associated arguments for each Block. Each argument may be a positive integer constant or any other Standard Numerical Attribute (SNA). A list of SNA's and their associated numerical codes is provided in Figure 7. SNA's have the same meaning here as they do in GPSS.

Each X-Vector element assigned to a Block entity contains two pieces of information, with the exception of an argument which is an

integer constant. Each piece of information is represented as a 2-byte integer value. The 1st element of a Block definition contains the Block type number in the first half of the element and the IPD pointer in the second half. The first half of the element of a GENERATE block would have a 1 in it, an ADVANCE block a 2, an ASSIGN block a 4 and so on. This value is used as the index number in a "Computed GOTO" statement to execute the routine associated with the block type and corresponds to the "Execution Routine Mask" in GPSS.

If there are 1 or more arguments associated with the Block being defined, then the next element will contain argument A, the following element will contain argument B, and so forth. If the argument is a positive integer constant, then the whole element will contain the constant. All other SNA's are represented in two parts. The first half of the element contains the negative of the SNA name code listed in Figure 7. The negative value is used to distinguish between constants and other SNA's. The SNA name code is used as an index in a "Computed GOTO" statement to execute the evaluation routine associated with that SNA name. The second half of the element is the entity number, if it is positive. If the second half is negative, indirect addressing is indicated, i.e., the absolute value of the second half of the element is the parameter number of the current Transaction which contains the entity number of the SNA. The following examples

show GPSS Blocks and arguments and the equivalent coding required for the Simulation Routine.

GPSS	X-Vector Block Representation	
GENERATE V1,2	1	
	-24	1
		2
	-33	0

ADVANCE 6, V*4	2	
		6
	-24	-4
	-33	0

The first part of the last element of a Block definition must be a -33 to signify to the Simulation Routine that the remaining arguments for this Block are zero. The second half of the last element is either zero or contains a numerical mode code which is used to define the exact operation required for a TEST, GATE, OR SELECT block. In GPSS this information is given as a modifier on the Block name. A table of mode codes appears in Figure 8. The Block routine uses this code as an index to a "Computer GOTO" statement within the routine associated with the Block type. The following examples illustrate the use of the mode code.

GPSS	X-Vector Block Representation	
TEST Ⓔ 5, Q1, 6	11	
		5
	-12	1
		6
	-33	1

GPSS	X-Vector Block Representation	
GATE SNF 2, P*3	16	
		2
	-1	-3
	-33	4
SELECT MIN 4, 2, 5, , S, 10	21	
		4
		2
		5
		0
	-9	0
		10
	-33	8

E. TABLES

Each Table entity requires a basic X-Vector element assignment of 9 contiguous elements. In addition, one element is required for each frequency class. All elements correspond to the Table allocation in GPSS. Figure 9 is a general layout of a Table definition in the X-Vector. One major difference between the Simulation Routine and GPSS is that in the X-Vector the SNA to be tabulated appears as argument A of a TABULATE block instead of being included in the Table definition. The calling program must initialize element 6 to the width of the frequency classes, element 8 to the number of frequency classes, element 9 to the upper limit of the lowest interval and all the other elements to zero.

The Simulation Routine will modify the elements each time a Transaction enters a TABULATE block which references the TABLE in Argument B. Sum of Arguments entered in Table is a signed

floating point value that is incremented by each argument value that is entered in the Table. The sum is used in the computation of the mean argument. Sum of squared values of the arguments entered in the Table is a floating point value that is incremented by the square of each argument value that is entered in the Table. The sum is used in the computation of the standard deviation of the Table argument. Sum of weighted value of arguments is a floating point value that is incremented by the product of the argument and the number of units specified by the argument C weighting factor of the TABULATE block. The sum is used in the computation of a weighted mean. Sum of weighted squares is a floating point value incremented by the product of the square of the argument and the number of units specified by the argument C weighting factor of the TABULATE block. The sum is used in the computation of a standard deviation of the weighted argument.

Number of entries in Table is incremented each time a Transaction enters a TABULATE block referencing the Table. The magnitude of the increment is specified by argument C of the TABULATE block. Sum of overflow arguments is a floating point value that is incremented by each argument value which lies in the last frequency interval of the Table. This sum can be divided by the number of entities in the last interval to obtain "Average Value of Overflow". A Frequency Interval corresponding to the argument is incremented by the value of argument C of the TABULATE block.

F. FUNCTIONS

Each Function entity defined in the model must be assigned 6 contiguous elements for basic information, plus additional contiguous elements for independent variable (X) values if any, Function (Y) values, and slope values, if any. Figure 10 is a general layout of a Function as it would appear in the X-Vector.

The Argument A SNA must be initialized by the calling program to the SNA that will be used as the search value. It can be any SNA allowed by the Simulation Routine, as listed in Figure 7. The next element must be initialized to a -23 in the first half and the IPD pointer in the second half. The third element should be initialized to zero. The first half of element 4 must be initialized to a -25. These last two elements are used by the simulation routine to evaluate the function values when they are not simply constants (i.e., when the mode is E or M).

Argument B, Part 1 (Mode) and Part 2 (Number of Points) must be initialized by the calling program in the fifth element. These values have the same meaning as in GPSS. The mode is a numeric code which is related to the GPSS mode as indicated in the following:

GPSS	Argument B-Part 1 (Mode)
C	1
D	2
E	3
L	4
M	5

Part 2 of this element is the number of points which define the Function.

The next element (the sixth) is initialized by the calling program. Part 1 (X value Integer/Decimal Indicator) must be set equal to 0 if the X values are integer and 1 if the X values are floating point. Part 2 (Y value Integer/Decimal Indicator) must be set equal to 0 if the Y values are integer and 1 if the Y values are floating point.

The X values and Y values must be initialized by the calling program to the values which define the Function. If the Function mode is continuous (Mode 1) then the calling program must reserve elements immediately following the Y values for slopes. The number of elements required is equal to the number of Function points. If the Function mode is discrete numerical or discrete attribute (Mode 2 or 3), then no elements are required for the slopes. If the Function is a list numerical or list attribute (Mode 4 or 5), then no elements are required for slopes or X values, since no search is performed. For these list mode Functions, the Y values must begin immediately in the seventh element. In those cases where slopes are needed they are computed in floating point by the Simulation Routine. The following examples illustrate the X-Vector representation of various GPSS Function definitions.

GPSS

X-Vector Function Representation

FUNCTION RN1, C2
0.3, 4.0/1.0, 11.0

-26	1
-23	
	0
-25	0
1	2
1	1
	0.3
	1.0
	4.0
	11.0
	0.0
	10.0

FUNCTION P3, D2
6, 4/8, 10

-1	3
-23	
	0
-25	0
2	2
0	0
	6
	8
	4
	10

FUNCTION V*4, M2
1, Q*1/2, Q*2

-24	-4
-23	
	0
-25	0
5	2
0	0
-12	-1
-12	-2

GPSS

```
FUNCTION RN3,E3
0.0,V*1/.542,15739688/
1.0,V10
```

X-Vector Function Representation

-26	3
-23	
	0
-25	0
3	3
1	0
	0.0
	0.542
	1.0
-24	-1
	15739688
-24	10

G. VARIABLES

Each Variable defined in the model must be completely specified by the calling program. A Variable definition requires contiguous X-Vector elements, with the number of elements depending on the complexity of the Variable. Each Variable is defined in "Polish post-fix" notation. The operand elements may be any SNA, including an integer constant, in the format discussed earlier. The operator elements must have a -30 in the first half and an operation code in the second half. This code is used to indicate which arithmetic operation is to be performed on the previous two evaluated arguments. The arithmetic codes are listed below.

Code	Arithmetic Operation
1	Add
2	Subtract
3	Multiply
4	Divide
5	Divide Modulo

The last element of a Variable definition must be initialized to a -25 in the first half. The following example shows two different ways in which the same Variable could be defined.

GPSS	X-Vector Variable Definitions				
VARIABLE 16+4*FN2	-22	2			16
		4			4
	-30	3	or	-22	2
		16		-30	3
	-30	1		-30	1
	-25	0		-25	0

H. TRANSACTIONS

Each Transaction entity requires a number of contiguous X-Vector elements which is equal to 8 plus the number of parameters assigned each Transaction (according to the value of X24). Figure 11 shows the layout of a Transaction entity. The calling program need not initialize Transactions in any way; the Simulation Routine will make all entries in these elements during the simulation.

When a Transaction is in the FEC, its first element contains the location of the previous Transaction in the FEC in the first half and the location of the next Transaction in the FEC in the second half. When a Transaction is in the CEC, the first half of this element contains the delay chain address of the next Transaction in the Chain, and the second half contains the current Block number. The second element contains similar values when the Transaction is in the FEC, except that there is no Transaction which is in the FEC and in a delay chain at the same time. The third element contains the clock time that

the Transaction is scheduled to leave the FEC. Each time the Transaction enters an ADVANCE block, this element will be set to some future time if the time delay is positive. The fourth element is the Absolute Clock time which is assigned each time the Transaction passes through a MARK block.

The first half of the fifth element contains the number of the next Block that the Transaction is scheduled to enter and the second half contains the priority. The sixth element has two logical indicators in the first half and a Queue number, when the Transaction is in a Queue, in the second half. The first indicator is not currently used and the second is in the scan status indicator, which is used by the Scan routine to determine if the Transaction is eligible for attempting to move it into its next Block. The next element is the time the Transaction entered the Queue whose number is in element 6. The eighth element contains the IPD pointer in the first half and the Transaction number in the second half. Transactions are numbered consecutively as they are created in GENERATE blocks. The remaining elements are the Transaction parameters, which are used in the same manner as in GPSS.

I. A SAMPLE PROBLEM

A sample problem is presented in Appendix A to show a complete X-Vector, both before and after execution of the routine. In the Appendix an equivalent GPSS program is given first. Then there is a

listing of the cards used to initialize the X-Vector, followed by a print-out of the X-Vector, produced by the Simulation Routine after execution. Finally there is a listing of the short FORTRAN program used to initialize the X-Vector and call the Simulation Routine.

The input listing can be seen to consist of four columns of numbers. Column 1 is the X-Vector subscript of the element to be loaded. Column 2 is 0 if the information is 4-byte integer, 1 if the information is two 2-byte integers and 2 if the information is a floating-point value. When column 2 equals 1, column 3 contains the first 2-byte integer. When column 2 equals 0 or 2, column 4 contains the 4-byte integer or floating-point value.

The X-Vector print-out can also be seen to consist of four columns of numbers. Column 1 is the X-Vector subscript of the element, column 2 is its hexadecimal representation, and columns 3 and 4 are the two 2-byte integer halves of the element. When the element has a floating-point value, it is printed in decimal form with an exponent.

III. THE PROCEDURE

The Simulation Routine is a FORTRAN subprogram which performs a simulation by manipulating information in the X-Vector. The algorithm which it follows is essentially the same as that of GPSS, with the X-Vector serving the same role as the "internal tables" of GPSS. The Simulation Routine consists logically of several routines, but physically it is just one subroutine (except for the Error handling routine).

In this section of the thesis the overall flow of the Simulation Routine will be described first, followed by a fairly detailed discussion of each of its parts: Scan and Move Transaction, Argument Evaluation, Block Routines, and Update Clock. Finally, there are discussions of tracing and error handling. A listing of the FORTRAN program is included at the end of the thesis, with a dictionary of most of the program variables appearing in Appendix B.

A. OVERALL FLOW OF ROUTINE

When the Simulation Routine is given control, it will perform the appropriate initialization of model entities according to the mode specified in X(1). Control is then given to the Scan routine which will attempt to move each Transaction on the CEC through as many Blocks as possible. Each time a Transaction attempts to enter a Block, the Block arguments will be evaluated and the Block routine performed.

When no Transaction can move into some next Block, control is given to the Update Clock routine which will increment the relative clock to the next event time in the FEC. All Transactions with this new clock time that are in the FEC will then be moved to the CEC. Control will again be given to the Scan routine. This cycle which is illustrated by Figures 12 through 14, will continue until some event within the model halts the simulation or an error occurs.

B. SCAN AND MOVE TRANSACTION

When control is passed to this routine, all Transactions with an event time less than or equal to the relative clock (X11) will have been merged into the CEC. These Transactions are chained together in the CEC in descending order of priority, as in GPSS. The Scan routine will reset the status change flag (SCFLAG) to off and examine the first Transaction on the CEC. If the scan status indicator (SSIND) is true, then an attempt will be made to move the Transaction into its next Block. If the SSIND is false, then some blocking condition within the model has caused the Transaction to become ineligible for movement into some next block. This condition will persist until the blocking condition is removed and the SSIND set to true or the Simulation Routine is called using mode 3 (Clear/Start) in which case all Transactions in the model are destroyed.

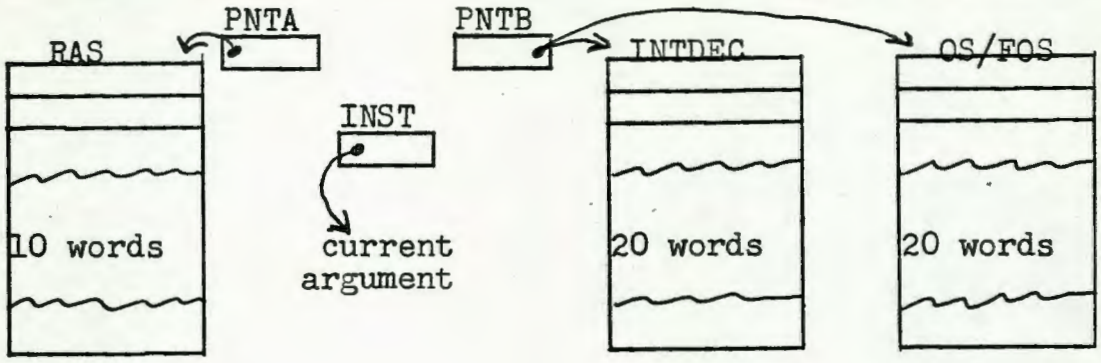
The Scan and Move Transaction routine illustrated in Figures 12 and 13 will move each Transaction on the CEC one at a time

through as many Blocks as possible. Each time a Block routine is executed, the SCFLAG is checked to see if there has been a status change, as in GPSS. If there has been a change, the SSIND for all appropriate Transactions is set to true. The Scan and Move Transaction routine will then continue to move the Transaction until it experiences a positive time delay in an ADVANCE block, enters a TERMINATE block or is blocked from moving into some next Block. If a positive time delay is encountered in an ADVANCE block, the Transaction is unlinked from the CEC and merged into the FEC in ascending order of Block departure time (BDT). Next the SCFLAG is checked. If it is on, the scan starts over with the first Transaction in the CEC. Otherwise the next sequential Transaction in the CEC is processed. When no Transaction can move into a next Block, control is transferred to the Update Clock routine.

C. EVALUATE ARGUMENTS

In section II it was stated that the arguments of a Block could be any SNA. These SNA codes may be considered to be instructions for a ficticious computer, indicating what routine must be performed to evaluate them. The following algorithm and flowchart describe in detail the argument evaluation procedure.

To evaluate SNA's requires 3 stacks and three pointers as indicated below.



As illustrated in the flowchart in Figure 15 the pointers PNTA and PNTB are initially set to zero and INST is set to the location (subscript) of the first argument of a Block. The fullword (4 byte) value of the argument is then checked for sign. If it is positive the whole argument is considered a positive integer constant and placed on the OS/FOS stack as a positive integer. If it is negative, the argument is either an SNA listed in Figure 7 or a code (-23, -25, -30, or -33).

All evaluated arguments will eventually appear in consecutive locations on the OS/FOS stack, and the corresponding integer/decimal indicator (INTDEC) elements will contain a 0 or 1 indicating whether the value on the OS/FOS stack is an integer or floating point value, respectively.

To place an evaluated argument on the OS/FOS stack, a routine must be performed which translates the SNA code to some value. The first thing which must be done is to check the second half of the

argument for a negative value which indicates indirect addressing. If the value is negative, the contents of the parameter, indicated by the absolute value of this number, contains the entity number portion of the SNA. This number is saved for later use. If the value of the second half is positive, it is the entity number to be used. The negative of the first half of the argument specifies the SNA name which is used as the index in a "Computed GOTO" statement.

When the "Computed GOTO" statement is executed, the appropriate routine produces a value for the SNA and places it on the top of the OS/FOS stack. The INST pointer is then incremented to point at the next Block argument. This procedure continues until a -33 code is detected. When the first half of the argument is a -33 code, this signals the end of arguments for this Block. The routine associated with code -33 determines whether a full seven arguments have been evaluated and placed on the OS/FOS stack. If not, this routine pads integer zeros into sequential OS/FOS and INTDEC locations after the last evaluated argument, up to and including the seventh location.

1. Variable Evaluation

The evaluation of a Variable is treated essentially as a recursive subroutine call upon the argument evaluation routine. The location of the next argument is saved on the return address stack (RAS), and then INST is set to the location of the first element of the Variable definition in the X-Vector. At this point argument evaluation continues with the operands of the Variable being evaluated and placed

on the operand stack (OS/FOS). When a Variable element is sensed with a -30 code in the first half, the arithmetic operation indicated by the code in the second half is performed on the last pair of evaluated operands on OS/FOS. The result then replaces these top two operands on the operand stack. When a Variable element is sensed with a -25 in the first half (at the end of the Variable definition), a return is made by popping the top return address off RAS and putting it in INST. The truncated value of the Variable is left on the top of OS/FOS as an integer. A Variable definition may contain other Variables. The only restriction is that a Variable must not have itself imbedded in its definition at any level.

2. Function Evaluation

When a Block argument has a -22 code in its first part, the Simulation Routine must depart from normal argument evaluation as it did in the case of a Variable. First the location of the next argument is placed on top of the return address stack (RAS). Then INST is set to the location of the 1st element of the Function definition which is the SNA to be used as the search argument. The argument evaluation routine then places the value of this SNA on top of the OS/FOS stack. The next argument to be evaluated will have a -23 code in the first half, which will cause the mode of the function to be determined. If the mode indicates that the Function is not "Attribute Valued", then the Function value is placed directly on the OS/FOS stack and a return is made to pop the location of the next argument to be evaluated off

the RAS stack. If the Function is "Attribute Valued" then the SNA to be returned as the Function value must be evaluated. By placing the SNA into the 3rd element of the Function definition, continuation of the argument evaluation routine will put the value of the SNA on top of the OS/FOS stack. The next argument will have a -25 code in the first half which will cause the Simulation Routine to return and pop the location of the next argument off RAS.

When locating the Function value to be returned, the same procedure is used for each mode as is done in GPSS. The search value and the Function values can be any SNA including another Function or a Variable. The Variable could have as its argument some other Function and so on. As long as the Function does not have itself imbedded in the search argument or in one of the Function values, there is essentially no limit to the depth to which this scheme can be carried.

D. BLOCK ROUTINES

After all arguments have been evaluated and their values placed on the OS/FOS stack, the Simulation Routine will execute a routine which corresponds to one of the Block type codes which are listed in Figure 6. Each routine is a section of FORTRAN code designed to perform the same function as the Block routines in GPSS. These routines have different arguments as listed in Figure 6, and each routine looks in particular elements of the OS/FOS stack for its

corresponding evaluated arguments. Normally, the Transaction being processed will proceed to the next sequential Block, except that certain Block routines may cause the Transaction to be delayed or transferred to some none sequential Block.

E. UPDATE CLOCK

As illustrated in Figures 12 and 14 this routine is executed when Transactions on the CEC can no longer move into some next Block. The relative clock (X11) is updated to the next most imminent event on the FEC, as in GPSS. All Transactions on the FEC with this new Block departure time (BDT) will then be moved to the CEC.

As Transactions are moved to the CEC, they are merged into the chain by priority. The Transactions are chained on the CEC in descending priority, i.e., highest priority number first. Each new Transaction becomes the last Transaction on the chain with that priority number.

After all Transactions have been moved, control is returned to the Scan routine.

F. TRACING VARIABLES

The Simulation Routine has provisions for tracing the execution of the simulation. Through the use of the NAMELIST feature in FORTRAN, it is possible to set the logical trace variables and vary the amount of tracing output. Trace variable TR6 is common to both the Simulation Routine and the calling program. If TR6, which

has a default value of "false", is set to "true", then the amount of tracing output may be specified by the user. With TR6 set to "true", the Simulation Routine will request the user to specify the logical value of various trace switches which control the tracing output.

G. ERROR CODES

Figure 2 is a list of codes for errors detectable by the Simulation Routine. When an error is detected, a subroutine is called. The arguments provided to the subroutine are an error number from Figure 2, a recovery point number (presently being used to locate the code where the error occurred since some errors can occur in more than one place), and the statement number to which the error subroutine is to return. The recovery number could be used in future expansion of the Simulation Routine to provide a link for continuing a simulation after an error has been corrected.

V. CONCLUSIONS AND RECOMMENDATIONS

A GPSS-like simulator callable from a FORTRAN program has been developed. This subroutine can be used by any program which properly initializes the X-Vector with information about the model, in a format similar to the "internal tables" of GPSS. Although only a portion of the complete GPSS capability was implemented, both the information structure and the Simulation Routine are such that expansion is straightforward.

It is recommended that work be done at this time to incorporate the simulator into the system described in Reference 1 for performing simulation analyses through natural language interaction with the computer. What is required essentially is a set of "encoding rules" which will initialize the X-Vector from an Internal Problem Description, call the Simulation Routine, and then report the results of the simulation. As the overall system is expanded to handle additional types of problems, the capabilities of the simulator can also be expanded accordingly.

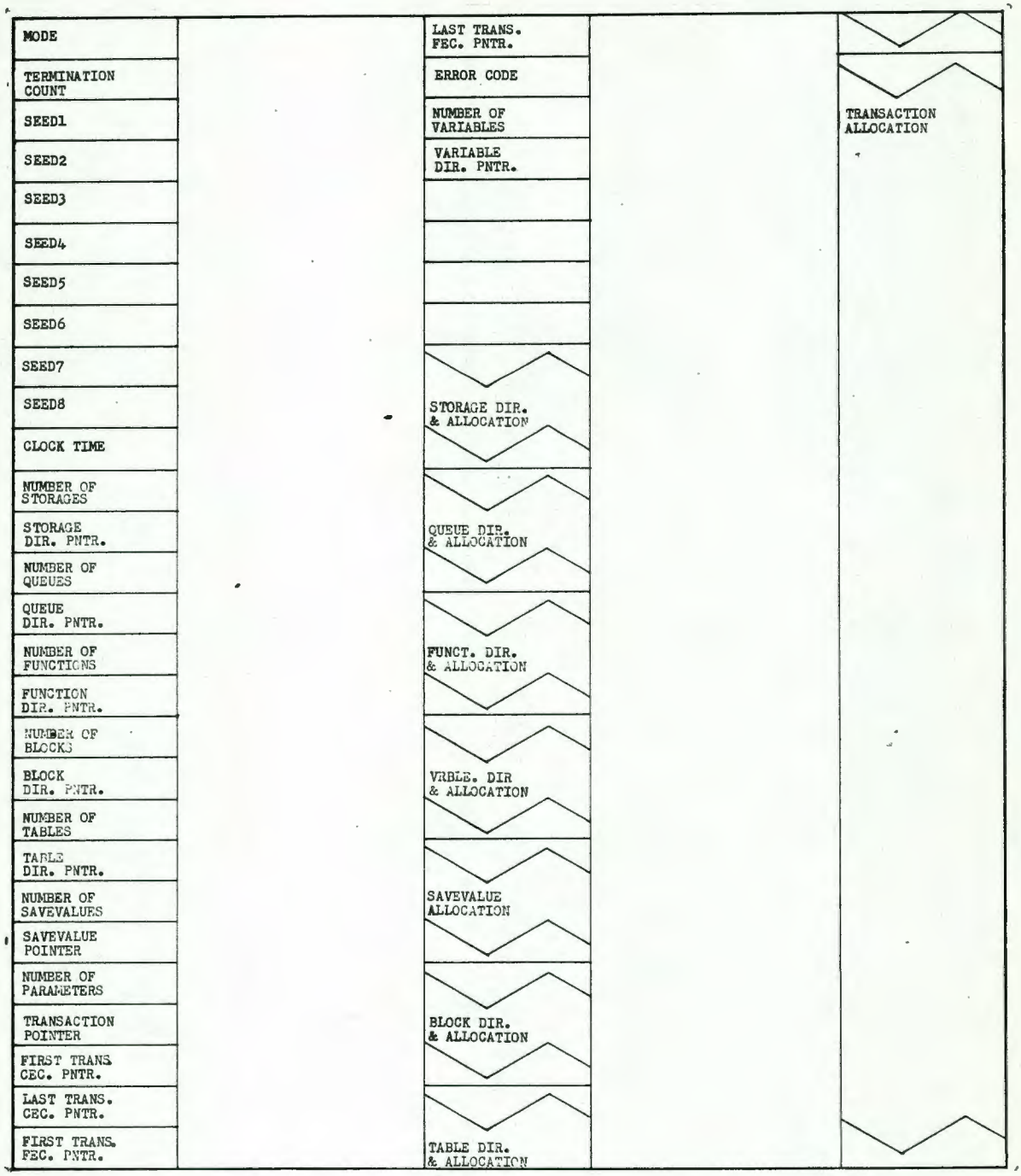


Figure 1 X-VECTOR GENERAL LAYOUT

CODE	MEANING
2	Illegal Block type code
45	Illegal SNA
401	No new event in system
425	Transaction leaving Storage by more than Storage contents
428	Transaction leaving Queue by more than Queue contents
433	Illegal Savevalue number
435	Illegal Table number
468	Number of Transactions exceeded
492	Illegal Transaction parameter number
499	Illegal Storage number
500	Illegal Queue number
505	Negative time delay computed
507	Illegal Function number
509	Illegal index evaluated for list-type Function
514	Illegal Variable number
603	Illegal Block number

Figure 2 ERROR CODES

CURRENT CONTENTS	
NUMBER OF AVAILABLE UNITS	
CUMULATIVE TIME INTEGRAL	
LAST CLOCK TIME STORAGE CHANGED STATUS	
ENTRY COUNT	
MAXIMUM STORAGE CONTENTS	
DELAY CHAIN ORIGIN FOR STORAGE FULL	DELAY CHAIN ORIGIN FOR STORAGE NOT EMPTY
DELAY CHAIN ORIGIN FOR STORAGE EMPTY	DELAY CHAIN ORIGIN FOR STORAGE NOT FULL
DELAY CHAIN ORIGIN FOR REDUCTION IN CONTENTS	IPD POINTER

Figure 3 STORAGE ENTITY LAYOUT

LAST CLOCK TIME QUEUE CHANGED STATUS	
TOTAL ENTRIES	
NUMBER OF ZERO DELAY ENTRIES	
CUMULATIVE TIME INTEGRAL	
CURRENT CONTENTS	
MAXIMUM CONTENTS	
IPD POINTER	BLANK

Figure 4 QUEUE ENTITY LAYOUT

BLOCK CODE	IPD POINTER
SNA NAME	ENTITY NUMBER
SNA NAME	ENTITY NUMBER
SNA NAME	ENTITY NUMBER
SNA NAME	ENTITY NUMBER
SNA NAME	ENTITY NUMBER
SNA NAME	ENTITY NUMBER
SNA NAME	ENTITY NUMBER
RETURN CODE -33	MODE CODE IF APPLICABLE OTHERWISE ZERO

Figure 5 BLOCK ENTITY LAYOUT

BLOCK TYPE CODE	BLOCK NAME AND ARGUMENTS
1	GENERATE A-Mean time B-Spread/Function modifier C-Initialization interval D-Creation limit E-Priority level
2	ADVANCE A-Mean time B-Spread/Function Modifier
3	TRANSFER B-Next Block
4	* ASSIGN A-Parameter number B-SNA to be assigned * C-Function modifier
5	MARK A-Parameter number
6	ENTER A-Index number of Storage B-Number of units to enter Storage
7	LEAVE A-Index number of Storage B-Number of units to leave Storage
8	QUEUE A-Index number of Queue B-Number of units to enter Queue
9	DEPART A-Index number of Queue B-Number of units to leave Queue
10	TERMINATE A-Termination count
11	* TEST A-First SNA B-Second SNA C-Next Block for a false relation

Figure 6 BLOCK TYPE CODES AND BLOCK ARGUMENTS

12	TABULATE * A-SNA to be tabulated * B-Index number of Table * C-Weighting factor
13	* SAVEVALUE A-Savevalue number B-SNA to be saved
16	* GATE A-Index number of Storage B-Next Block if false condition
21	* SELECT A-Parameter number B-Lower limit of entity class C-Upper limit of entity class D-Comparison value, if applicable E-Entity to be examined F-Next Block if condition not met

*The Blocks and Arguments marked differ slightly from their GPSS counterparts, as discussed in Section II-D

Figure 6 continued

STANDARD NUMERICAL ATTRIBUTE (SNA) NAME	CODE
Parameter (P)	1
Priority (PR)	2
Transit Time (M)	3
Parameter Transit Time (MP)	4
Storage Entry Count (SC)	5
Storage Maximum Contents (SM)	6
Storage Average Utilization (SR)	7
Storage Average Contents (SA)	8
Storage Current Contents (S)	9
Storage Remaining Contents (R)	10
Storage Average Time per Transaction (ST)	11
Queue Current Length (Q)	12
Queue Average Contents (QA)	13
Queue Maximum Contents (QM)	14
Queue Entry Count (QC)	15
Queue Zero Entries (QZ)	16
Queue Average Time per Transaction (QT)	17
Queue Average Time per Transaction excluding Zero (QX)	18
Table Mean (TB)	19
Table Entry Count (TC)	20
Table Standard Deviation (TD)	21
Function (FN)	22

Figure 7 STANDARD NUMERICAL ATTRIBUTE (SNA) CODES

Variable (V)	24
Random Number (RN)	26
Clock Time (C)	27
Savevalue (X)	28

Figure 7 continued

CODE	SELECT	TEST	GATE	ASSIGN/SAVEVALUE
1	E	E	SNE	Subtraction
2	L	L	SF	Replacement
3	LE	LE	SNF	Addition
4	GT	GT	SE	
5	GE	GE		
6	U	U		
7	MAX			
8	MIN			
9	SNE			
10	SF			
11	SNF			
12	SE			

Figure 8 MODE CODES

SUM OF ARGUMENTS ENTERED IN TABLE
SUM OF SQUARES OF ARGUMENTS ENTERED IN TABLE
SUM OF WEIGHTED VALUES OF ARGUMENTS
SUM OF WEIGHTED SQUARES
NUMBER OF ENTRIES IN TABLE
WIDTH OF TABLE FREQUENCY CLASSES
SUM OF OVERFLOW ARGUMENTS
NUMBER OF FREQUENCY CLASSES IN TABLE
UPPER LIMIT OF LOWEST INTERVAL
FREQUENCY INTERVALS

Figure 9 TABLE ENTITY LAYOUT

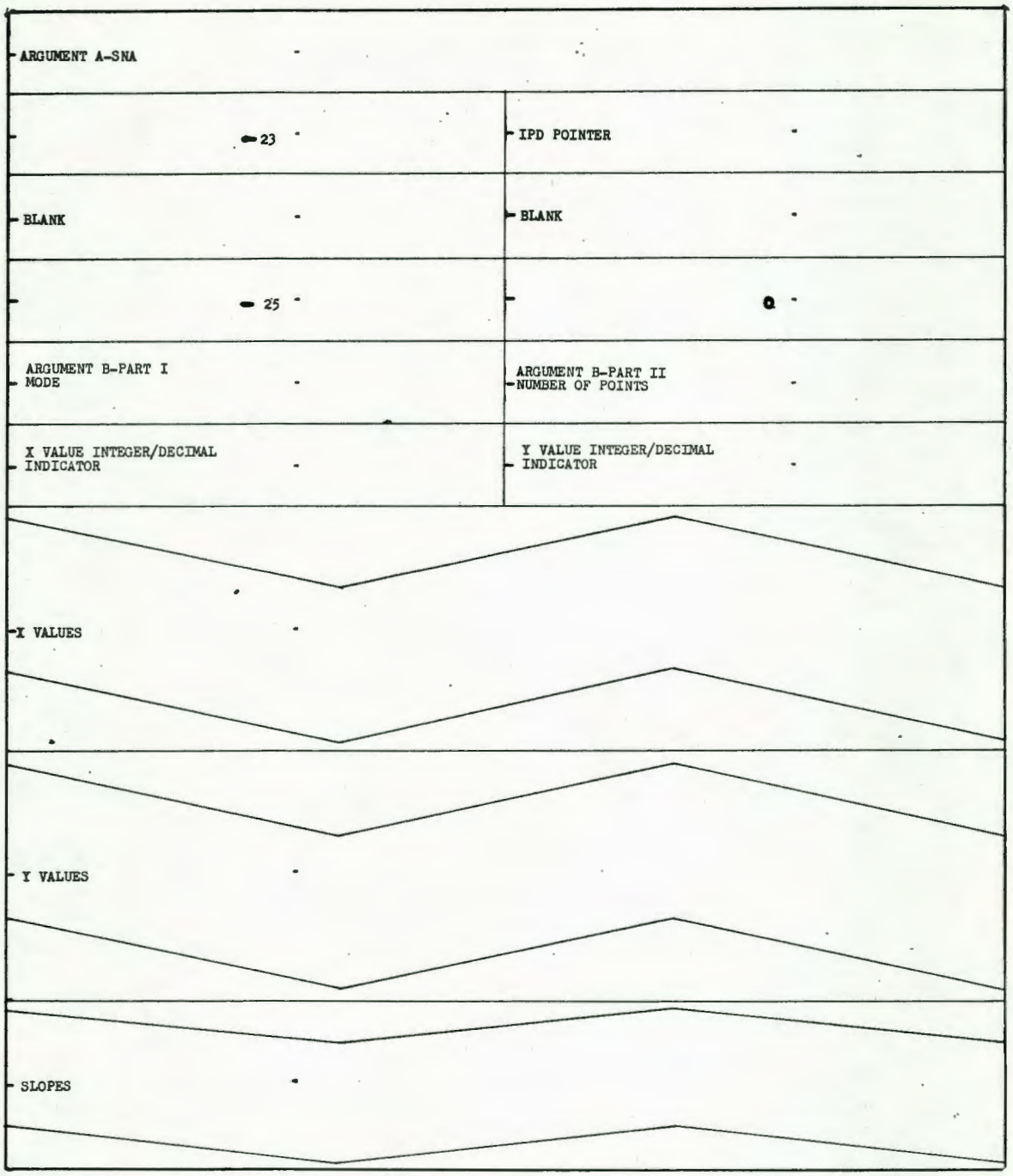


Figure 10 FUNCTION ENTITY LAYOUT

PREV TRANS IN FEC (DELAY CHAIN ADDRESS OF NEXT TRANS IN CHAIN)		NEXT TRANS IN FEC (CURRENT BLOCK NUMBER WHEN TRANS IN CEC)
PREVIOUS TRANSACTION IN CURRENT EVENTS CHAIN		NEXT TRANS CEC (CURRENT BLOCK NUMBER WHEN TRANS IN FEC)
TIME TRANSACTION SCHEDULED TO LEAVE FUTURE EVENTS CHAIN		
MARK TIME		
NEXT BLOCK FOR TRANSACTION		PRIORITY
TRACE INDICATOR	SCAN INDICATOR	QUEUE NUMBER
TIME TRANSACTION ENTERED QUEUE		
IPD POINTER		TRANSACTION NUMBER
PARAMETER 1		
PARAMETER 2		
PARAMETER N		

Figure 11 TRANSACTION ENTITY LAYOUT

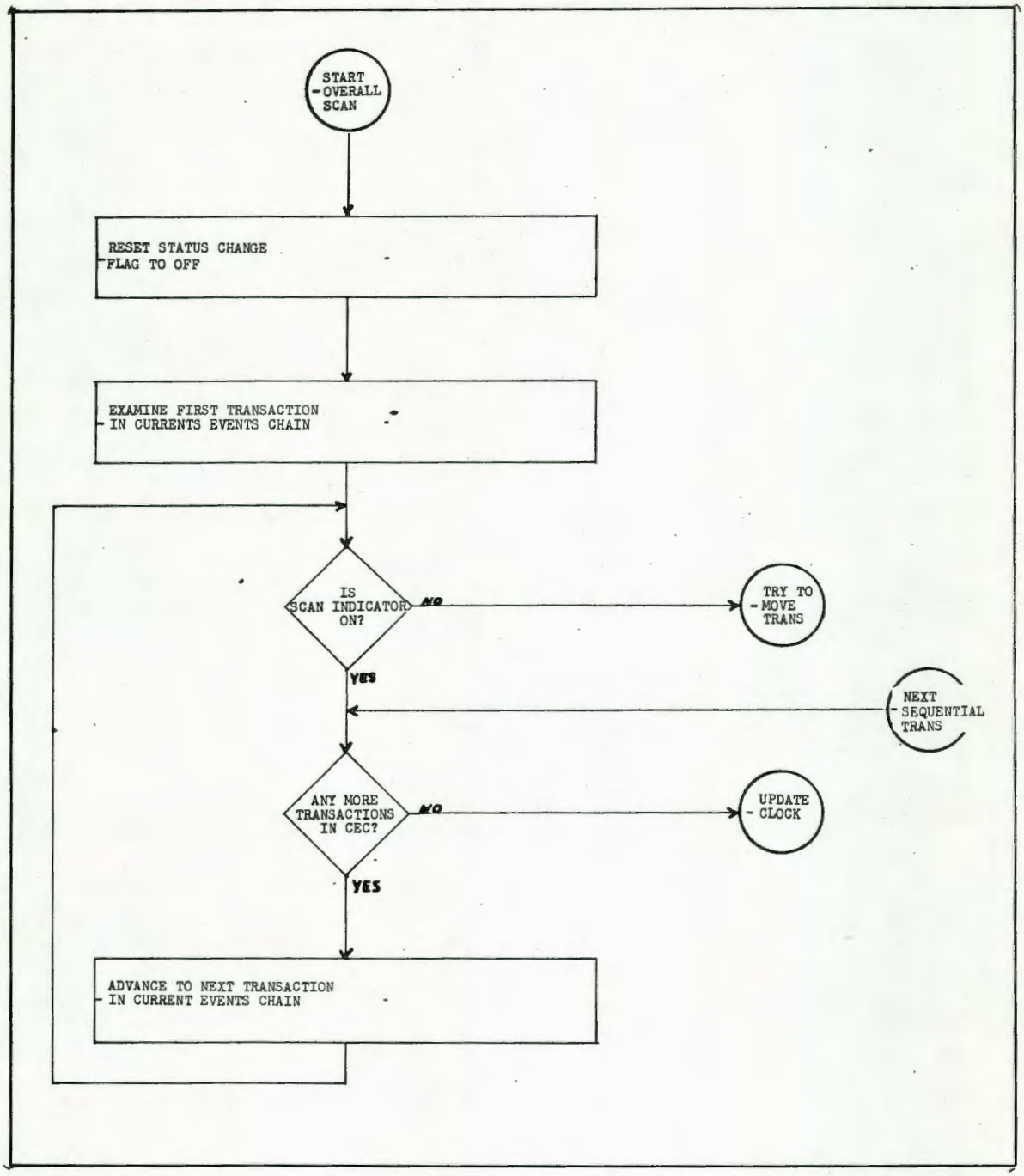


Figure 12 FLOWCHART OF SCAN ROUTINE

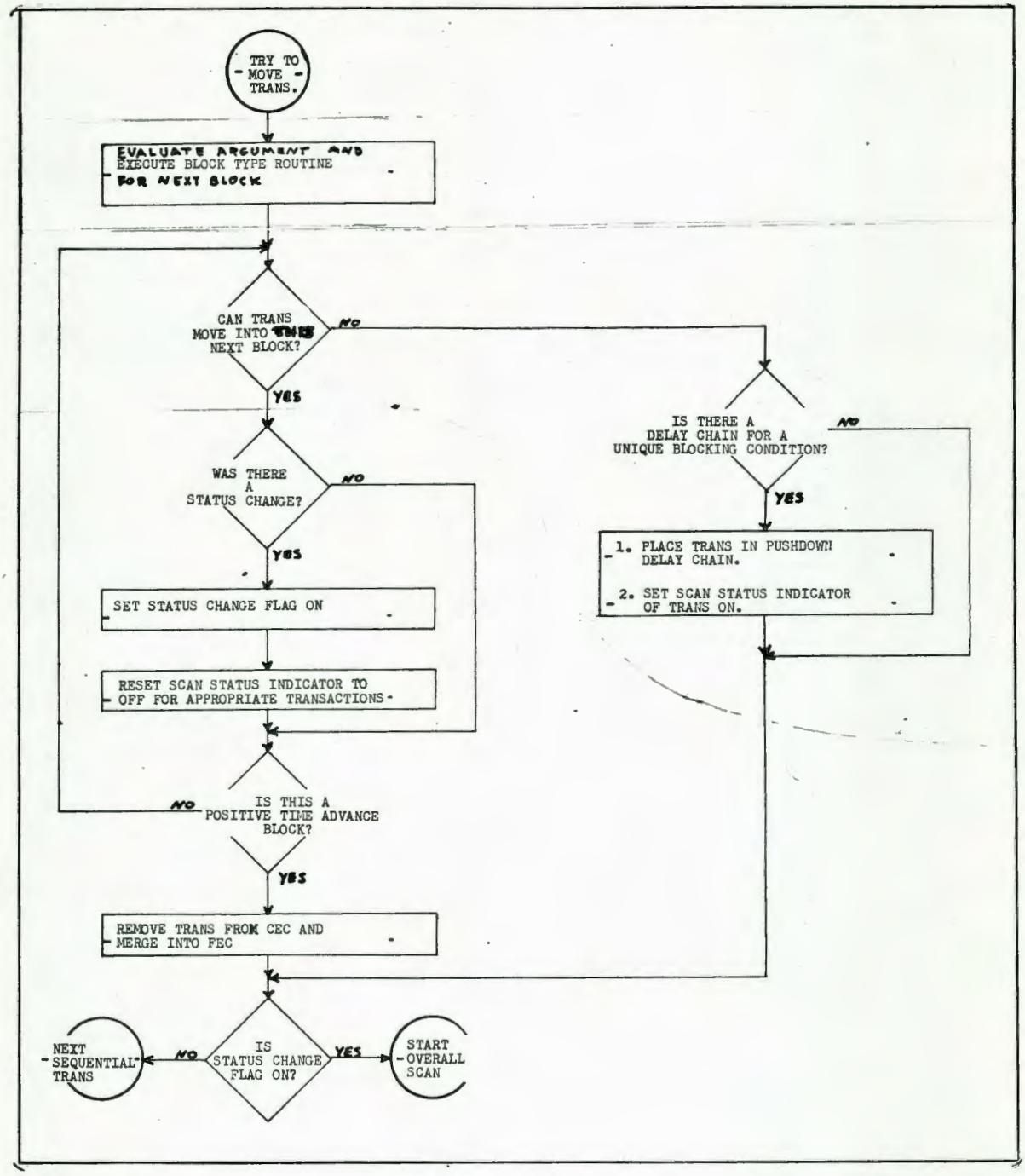


Figure 13 FLOWCHART OF MOVE TRANSACTION ROUTINE

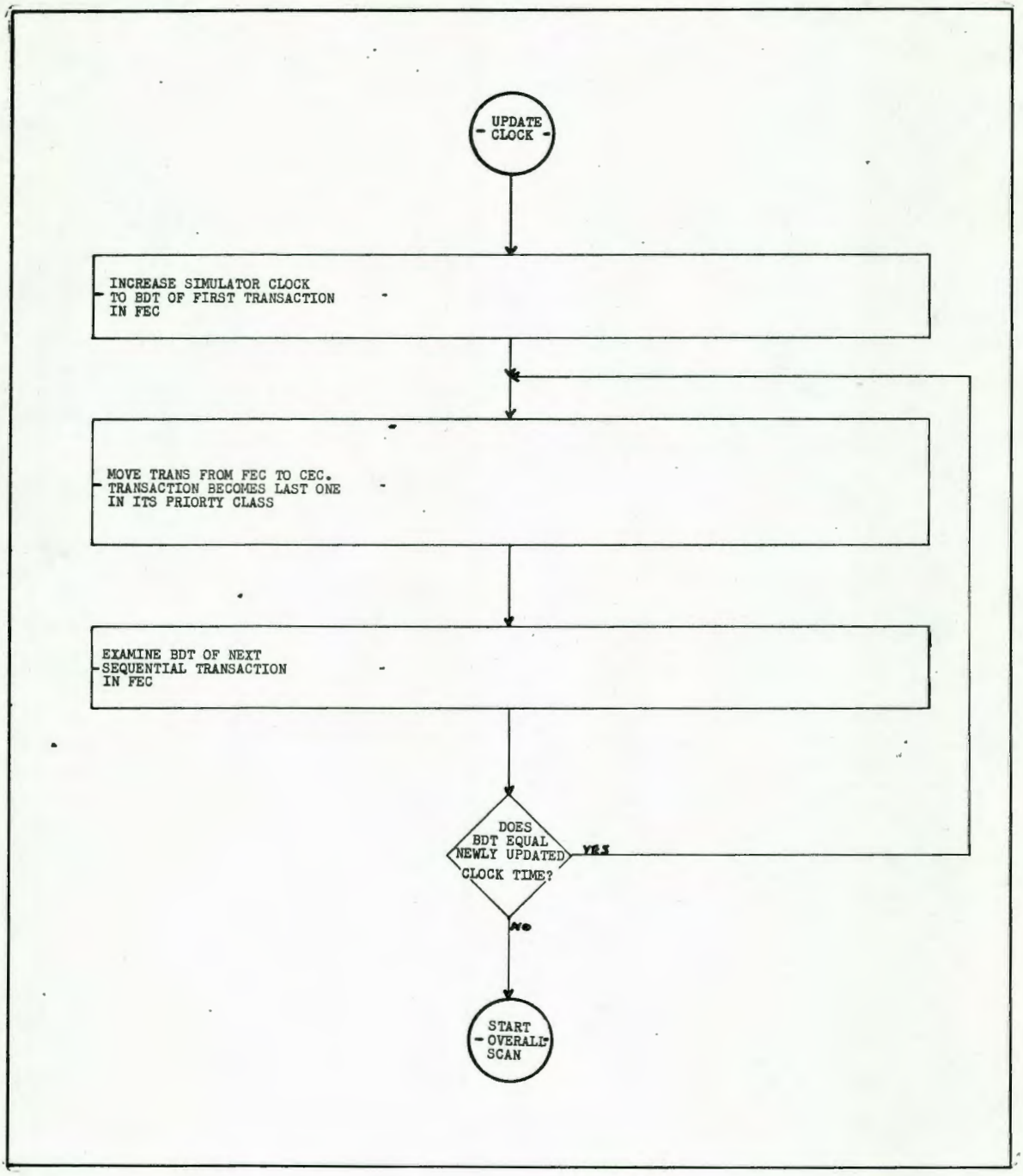


Figure 14 FLOWCHART OF UPDATE CLOCK ROUTINE

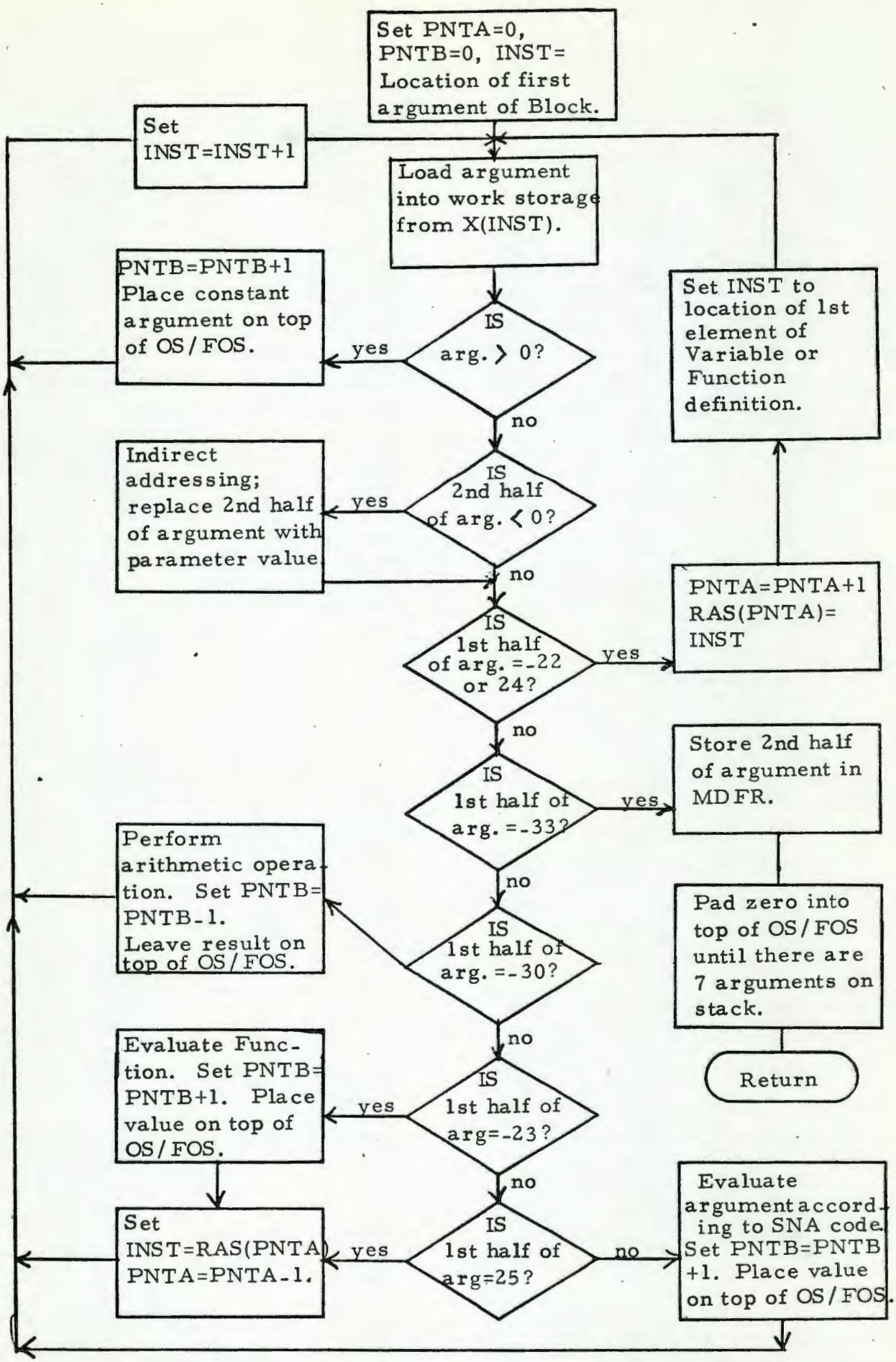


Figure 15 FLOWCHART OF ARGUMENT EVALUATION ROUTINE

APPENDIX A

SAMPLE PROBLEM

```
        SIMULATE
        RMULT      277,423,715,121,655,531,999,813
 2      TABLE      M1,,1,2
 3      TABLE      M1,,1,2
 2      STORAGE    1
 1      FUNCTION   RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/
.75,1.39/.8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/
.95,2.99/.96,3.2/.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/
.999,7/.9997,8/
 2      FUNCTION   RN2,C29
0,-3/.012,-2.25/.027,-1.93/.043,-1.72/.062,-1.54/.084,-1.38/
.104,-1.26/.131,-1.12/.159,-1/.187,-.89/.23,-.74/.267,-.62/
.334,-.43/.432,-.17/.5,0/.568,.17/.666,.43/.732,.62/.77,.74/
.813,.89/.841,1/.988,2.25/1,3/
 3      FUNCTION   P1,D2
2,10/3,18/
 4      FUNCTION   RN3,D2
.750,2/1.000,3/
 1      FVARIABLE  16+4*FN2

 1      GENERATE   ~~V1~~ 37
 2      ASSIGN     1,FN4
 3      TEST L     Q2,2,5
 4      TRANSFER   ,7

 5      TABULATE   P1
 6      TERMINATE

 7      QUEUE      2
 8      ENTER      2
 9      DEPART     2
 10     ADVANCE    FN3,FN1
 11     LEAVE      2
 12     TRANSFER   ,5

 13     GENERATE   180
 14     TERMINATE  1
        START      1
        END
```

INPUT FILE WITH X-VECTOREQUIVALENT OF SAMPLE PROBLEM

360			
1	0	0	1
2	0	0	1
3	0	0	277
4	0	0	423
5	0	0	715
6	0	0	121
7	0	0	655
8	0	0	531
9	0	0	999
10	0	0	813
11	0	0	0
12	0	0	2
13	0	0	32
14	0	0	2
15	0	0	54
16	0	0	4
17	0	0	140
18	0	0	14
19	0	0	72
20	0	0	3
21	0	0	335
22	0	0	0
23	0	0	0
24	0	0	12
25	0	0	360
26	0	0	0
27	0	0	0
28	0	0	0
29	0	0	0
30	0	0	0
31	0	0	1
32	0	0	133
33	0	0	34
34	0	0	44
35	0	0	0
36	0	0	1
37	0	0	0
38	0	0	0
39	0	0	0
40	0	0	0
41	0	0	0
42	0	0	0
43	0	0	0
44	0	0	0
45	0	0	0
46	0	0	1
47	0	0	0
48	0	0	0
49	0	0	0
50	0	0	0
51	0	0	0
52	0	0	0
53	0	0	0
54	0	0	0
55	0	0	56
56	0	0	64
57	0	0	0
58	0	0	0
59	0	0	0
60	0	0	0
61	0	0	0
62	0	0	0
63	0	0	0
64	0	0	0
65	0	0	0
66	0	0	0
67	0	0	0
68	0	0	0
69	0	0	0

70	0	0	0
71	0	0	0
72	0	0	0
73	0	0	86
74	0	0	89
75	0	0	93
76	0	0	98
77	0	0	101
78	0	0	105
79	0	0	107
80	0	0	110
81	0	0	113
82	0	0	116
83	0	0	120
84	0	0	123
85	0	0	126
86	0	0	129
87	1	1	1
88	0	0	37
89	1	-33	0
90	1	4	1
91	0	0	1
92	1	-22	4
93	1	-33	2
94	1	11	1
95	1	-12	2
96	0	0	2
97	0	0	5
98	1	-33	2
99	1	3	1
100	0	0	7
101	1	-33	0
102	1	12	1
103	1	-3	1
104	1	-1	1
105	1	-33	0
106	1	10	1
107	1	-33	0
108	1	8	1
109	0	0	2
110	1	-33	0
111	1	6	1
112	0	0	2
113	1	-33	0
114	1	9	1
115	0	0	2
116	1	-33	0
117	1	2	1
118	1	-22	3
119	1	-22	1
120	1	-33	0
121	1	7	1
122	0	0	2
123	1	-33	0
124	1	3	1
125	0	0	5
126	1	-33	0
127	1	1	1
128	0	0	180
129	1	-33	0
130	1	10	1
131	0	0	1
132	1	-33	0
133	0	0	0
134	0	0	134
135	0	0	16
136	0	0	4
137	1	-22	2
138	1	-30	3
139	1	-30	1
140	1	-25	0
141	0	0	144

142	0	0	222
143	0	0	315
144	0	0	325
145	1	-26	1
146	1	-23	0
147	0	0	0
148	1	-25	0
149	1	1	24
150	1	1	1
151	2	0	0.0
152	2	0	0.1
153	2	0	0.2
154	2	0	0.3
155	2	0	0.4
156	2	0	0.5
157	2	0	0.6
158	2	0	0.7
159	2	0	0.75
160	2	0	0.8
161	2	0	0.84
162	2	0	0.88
163	2	0	0.9
164	2	0	0.92
165	2	0	0.94
166	2	0	0.95
167	2	0	0.96
168	2	0	0.97
169	2	0	0.98
170	2	0	0.99
171	2	0	0.995
172	2	0	0.998
173	2	0	0.999
174	2	0	0.9997
175	2	0	0.0
176	2	0	0.104
177	2	0	0.222
178	2	0	0.355
179	2	0	0.509
180	2	0	0.69
181	2	0	0.915
182	2	0	1.2
183	2	0	1.39
184	2	0	1.6
185	2	0	1.83
186	2	0	2.12
187	2	0	2.3
188	2	0	2.52
189	2	0	2.81
190	2	0	2.99
191	2	0	3.2
192	2	0	3.5
193	2	0	3.9
194	2	0	4.6
195	2	0	5.3
196	2	0	6.2
197	2	0	7.0
198	2	0	8.0
199	0	0	0
200	0	0	0
201	0	0	0
202	0	0	0
203	0	0	0
204	0	0	0
205	0	0	0
206	0	0	0
207	0	0	0
208	0	0	0
209	0	0	0
210	0	0	0
211	0	0	0
212	0	0	0
213	0	0	0

214	0	0	0
215	0	0	0
216	0	0	0
217	0	0	0
218	0	0	0
219	0	0	0
220	0	0	0
221	0	0	0
222	0	0	0
223	1	-26	2
224	1	-23	0
225	0	0	0
226	1	-25	0
227	1	1	29
228	1	1	1
229	2	0	0.0
230	2	0	.012
231	2	0	.027
232	2	0	.043
233	2	0	.062
234	2	0	.084
235	2	0	.104
236	2	0	.131
237	2	0	.159
238	2	0	.187
239	2	0	.23
240	2	0	.267
241	2	0	.334
242	2	0	.432
243	2	0	.5
244	2	0	.568
245	2	0	.666
246	2	0	.732
247	2	0	.77
248	2	0	.813
249	2	0	.841
250	2	0	.869
251	2	0	.896
252	2	0	.916
253	2	0	.938
254	2	0	.957
255	2	0	.973
256	2	0	.988
257	2	0	1.0
258	2	0	-3.0
259	2	0	-2.25
260	2	0	-1.93
261	2	0	-1.72
262	2	0	-1.54
263	2	0	-1.38
264	2	0	-1.26
265	2	0	-1.12
266	2	0	-1.0
267	2	0	-0.89
268	2	0	-0.74
269	2	0	-0.62
270	2	0	-0.43
271	2	0	-0.17
272	2	0	0.0
273	2	0	.17
274	2	0	.43
275	2	0	.63
276	2	0	.74
277	2	0	.89
278	2	0	1.0
279	2	0	1.12
280	2	0	1.26
281	2	0	1.38
282	2	0	1.54
283	2	0	1.72
284	2	0	1.93
285	2	0	2.25

286	2	0	3.0
287	0	0	0
288	0	0	0
289	0	0	0
290	0	0	0
291	0	0	0
292	0	0	0
293	0	0	0
294	0	0	0
295	0	0	0
296	0	0	0
297	0	0	0
298	0	0	0
299	0	0	0
300	0	0	0
301	0	0	0
302	0	0	0
303	0	0	0
304	0	0	0
305	0	0	0
306	0	0	0
307	0	0	0
308	0	0	0
309	0	0	0
310	0	0	0
311	0	0	0
312	0	0	0
313	0	0	0
314	0	0	0
315	0	0	0
316	1	-1	1
317	1	-23	0
318	0	0	0
319	1	-25	1
320	1	2	2
321	0	0	0
322	0	0	2
323	0	0	3
324	0	0	10
325	0	0	18
326	1	-26	3
327	1	-23	0
328	0	0	0
329	1	-25	0
330	1	2	2
331	1	1	0
332	2	0	0.750
333	2	0	1.000
334	0	0	2
335	0	0	3
336	0	0	500
337	0	0	338
338	0	0	349
339	0	0	0
340	0	0	0
341	0	0	0
342	0	0	0
343	0	0	0
344	0	0	1
345	0	0	0
346	0	0	2
347	0	0	0
348	0	0	0
349	0	0	0
350	0	0	0
351	0	0	0
352	0	0	0
353	0	0	0
354	0	0	0
355	0	0	1
356	0	0	0
357	0	0	2

358	0	0	0
359	0	0	0
360	0	0	0

X-VECTOR STATUS AT END OF SIMULATION RUN

	FW1 HEX	HW1 DEC	HW2 DEC	
1	00000001	0	1	
2	00000000	0	0	
3	74DC57A5	29916	22437	0.354077E 63
4	000001A7	0	423	
5	2DA4E23B	11684	-7621	
6	00000079	0	121	
7	0000028F	0	655	
8	00000213	0	531	
9	000003E7	0	999	
10	0000032D	0	813	
11	000000B4	0	180	
12	00000002	0	2	
13	00000020	0	32	
14	00000002	0	2	
15	00000036	0	54	
16	00000004	0	4	
17	0000008C	0	140	
18	0000000E	0	14	
19	00000048	0	72	
20	00000003	0	3	
21	0000014F	0	335	
22	00000000	0	0	
23	00000000	0	0	
24	0000000C	0	12	
25	00000168	0	360	
26	00000000	0	0	
27	00000000	0	0	
28	00000168	0	360	
29	00000168	0	360	
30	00000000	0	0	
31	00000001	0	1	
32	00000085	0	133	
33	00000022	0	34	
34	0000002C	0	44	
35	00000000	0	0	
36	00000001	0	1	
37	00000000	0	0	
38	00000000	0	0	
39	00000000	0	0	
40	00000000	0	0	
41	00000000	0	0	
42	00000000	0	0	
43	00000000	0	0	
44	00000000	0	0	
45	00000000	0	0	
46	00000001	0	1	
47	00000000	0	0	
48	00000000	0	0	
49	00000094	0	148	
50	00000004	0	4	
51	00000001	0	1	
52	00000000	0	0	
53	00000000	0	0	
54	00000000	0	0	
55	00000038	0	56	
56	00000040	0	64	
57	00000000	0	0	
58	00000000	0	0	
59	00000000	0	0	
60	00000000	0	0	
61	00000000	0	0	
62	00000000	0	0	
63	00000000	0	0	
64	00000000	0	0	
65	00000094	0	148	
66	00000004	0	4	
67	00000004	0	4	
68	00000000	0	0	

69	00000000	0	0
70	00000000	0	0
71	00000001	0	1
72	00000000	0	0
73	00000056	0	86
74	00000059	0	89
75	0000005D	0	93
76	00000062	0	98
77	00000065	0	101
78	00000069	0	105
79	0000006B	0	107
80	0000006E	0	110
81	00000071	0	113
82	00000074	0	116
83	00000078	0	120
84	0000007B	0	123
85	0000007E	0	126
86	00000081	0	129
87	00010001	1	1
88	00000025	0	37
89	FFDF0000	-33	0
90	00040001	4	1
91	00000001	0	1
92	FFEA0004	-22	4
93	FFDF0002	-33	2
94	000B0001	11	1
95	FFF40002	-12	2
96	00000002	0	2
97	00000005	0	5
98	FFDF0002	-33	2
99	00030001	3	1
100	00000007	0	7
101	FFDF0000	-33	0
102	000C0001	12	1
103	FFFD0001	-3	1
104	FFFF0001	-1	1
105	FFDF0000	-33	0
106	000A0001	10	1
107	FFDF0000	-33	0
108	00080001	8	1
109	00000002	0	2
110	FFDF0000	-33	0
111	00060001	6	1
112	00000002	0	2
113	FFDF0000	-33	0
114	00090001	9	1
115	00000002	0	2
116	FFDF0000	-33	0
117	00020001	2	1
118	FFEA0003	-22	3
119	FFEA0001	-22	1
120	FFDF0000	-33	0
121	00070001	7	1
122	00000002	0	2
123	FFDF0000	-33	0
124	00030001	3	1
125	00000005	0	5
126	FFDF0000	-33	0
127	00010001	1	1
128	000000B4	0	180
129	FFDF0000	-33	0
130	000A0001	10	1
131	00000001	0	1
132	FFDF0000	-33	0
133	00000000	0	0
134	00000086	0	134
135	00000010	0	16
136	00000004	0	4
137	FFEA0002	-22	2
138	FFE20003	-30	3
139	FFE20001	-30	1
140	FFE70000	-25	0

141	00000090	0	144	
142	000000DE	0	222	
143	00000138	0	315	
144	00000145	0	325	
145	FFE60001	-26	1	
146	FFE90000	-23	0	
147	00000000	0	0	
148	FFE70000	-25	0	
149	00010018	1	24	
150	00010001	1	1	
151	00000000	0	0	
152	40199999	16409	-26215	0.100000E 00
153	40333333	16435	13107	0.200000E 00
154	404CCCCC	16460	-13108	0.300000E 00
155	40666666	16486	26214	0.400000E 00
156	40800000	16512	0	0.500000E 00
157	40999999	16537	-26215	0.600000E 00
158	40B33333	16563	13107	0.700000E 00
159	40C00000	16576	0	0.750000E 00
160	40CCCCCC	16588	-13108	0.800000E 00
161	40D70A3D	16599	2621	0.840000E 00
162	40E147AE	16609	18350	0.880000E 00
163	40E66666	16614	26214	0.900000E 00
164	40EB851E	16619	-31458	0.920000E 00
165	40F0A3D7	16624	-23593	0.940000E 00
166	40F33333	16627	13107	0.950000E 00
167	40F5C28F	16629	-15729	0.960000E 00
168	40F851EB	16632	20971	0.970000E 00
169	40FAE147	16634	-7865	0.980000E 00
170	40FD70A3	16637	28835	0.990000E 00
171	40FEB851	16638	-18351	0.995000E 00
172	40FF7CED	16639	31981	0.998000E 00
173	40FFBE76	16639	-16778	0.999000E 00
174	40FFEC56	16639	-5034	0.999700E 00
175	00000000	0	0	
176	401A9FBE	16410	-24642	0.104000E 00
177	4038D4FD	16440	-11011	0.222000E 00
178	405AE147	16474	-7865	0.355000E 00
179	40824DD2	16514	19922	0.509000E 00
180	40B0A3D7	16560	-23593	0.690000E 00
181	40EA3D70	16618	15728	0.915000E 00
182	41133333	16659	13107	0.120000E 01
183	41163D70	16662	15728	0.139000E 01
184	41199999	16665	-26215	0.160000E 01
185	411D47AE	16669	18350	0.183000E 01
186	4121EB85	16673	-5243	0.212000E 01
187	4124CCCC	16676	-13108	0.230000E 01
188	412851EB	16680	20971	0.252000E 01
189	412CF5C2	16684	-2622	0.281000E 01
190	412FD70A	16687	-10486	0.299000E 01
191	41333333	16691	13107	0.320000E 01
192	41380000	16696	0	0.350000E 01
193	413E6666	16702	26214	0.390000E 01
194	41499999	16713	-26215	0.460000E 01
195	4154CCCC	16724	-13108	0.530000E 01
196	41633333	16739	13107	0.620000E 01
197	41700000	16752	0	0.700000E 01
198	41800000	16768	0	0.800000E 01
199	00000000	0	0	
200	4110A3D7	16656	-23593	0.104000E 01
201	4112E147	16658	-7865	0.118000E 01
202	411547AE	16661	18350	0.133000E 01
203	4118A3D6	16664	-23594	0.154000E 01
204	411CF5C2	16668	-2622	0.181000E 01
205	41240000	16676	0	0.225000E 01
206	412D9997	16685	-26217	0.285000E 01
207	413CCCC3	16700	-13117	0.379999E 01
208	41433338	16707	13112	0.420000E 01
209	415C00C9	16732	9	0.575001E 01
210	4173FFFA	16755	-6	0.724999E 01
211	418FFFE6	16783	-26	0.899998E 01
212	41B00019	16816	25	0.110000E 02

213	41E7FFDF	16871	-33	0.145000E 02
214	42120003	16914	3	0.180000E 02
215	42150001	16917	1	0.210000E 02
216	421E0003	16926	3	0.300000E 02
217	42280000	16936	0	0.400000E 02
218	42460003	16966	3	0.700000E 02
219	428C0006	17036	6	0.140000E 03
220	4312BFF8	17170	-16392	0.299998E 03
221	4332002A	17202	42	0.800010E 03
222	43594 93E	17241	18750	0.142858E 04
223	FFE60002	-26	2	
224	FFE90000	-23	0	
225	00000000	0	0	
226	FFE70000	-25	0	
227	0001001D	1	29	
228	00010001	1	1	
229	00000000	0	0	
230	3F3126E9	16177	9961	0.120000E-01
231	3F6E978D	16238	-26739	0.270000E-01
232	3FB020C4	16304	8388	0.430000E-01
233	3FFDF3B6	16381	-3146	0.620000E-01
234	40158106	16405	-32506	0.840000E-01
235	401A9FBE	16410	-24642	0.104000E 00
236	40218937	16417	-30409	0.131000E 00
237	4028B439	16424	-19399	0.159000E 00
238	402FDF3B	16431	-8389	0.187000E 00
239	403AE147	16442	-7865	0.230000E 00
240	40445A1C	16452	23068	0.267000E 00
241	40558106	16469	-32506	0.334000E 00
242	406E978D	16494	-26739	0.432000E 00
243	40800000	16512	0	0.500000E 00
244	40916872	16529	26738	0.568000E 00
245	40AA7EF9	16554	32505	0.666000E 00
246	40BB645A	16571	25690	0.732000E 00
247	40C51EB8	16581	7864	0.770000E 00
248	40D020C4	16592	8388	0.813000E 00
249	40D74BC6	16599	19398	0.841000E 00
250	40DE76C8	16606	30408	0.869000E 00
251	40E56041	16613	24641	0.896000E 00
252	40EA7EF9	16618	32505	0.916000E 00
253	40F020C4	16624	8388	0.938000E 00
254	40F4FDF3	16628	-525	0.957000E 00
255	40F91687	16633	5767	0.973000E 00
256	40FCED91	16636	-4719	0.988000E 00
257	41100000	16656	0	0.100000E 01
258	C1300000	-16080	0	-.300000E 01
259	C1240000	-16092	0	-.225000E 01
260	C11EE147	-16098	-7865	-.193000E 01
261	C11B851E	-16101	-31458	-.172000E 01
262	C118A3D7	-16104	-23593	-.154000E 01
263	C116147A	-16106	5242	-.138000E 01
264	C11428F5	-16108	10485	-.126000E 01
265	C111EB85	-16111	-5243	-.112000E 01
266	C1100000	-16112	0	-.100000E 01
267	C0E3D70A	-16157	-10486	-.890000E 00
268	C0BD70A3	-16195	28835	-.740000E 00
269	C09EB851	-16226	-18351	-.620000E 00
270	C06E147A	-16274	5242	-.430000E 00
271	C02B851E	-16341	-31458	-.170000E 00
272	00000000	0	0	
273	402B851E	16427	-31458	0.170000E 00
274	406E147A	16494	5242	0.430000E 00
275	40A147AE	16545	18350	0.630000E 00
276	40BD70A3	16573	28835	0.740000E 00
277	40E3D70A	16611	-10486	0.890000E 00
278	41100000	16656	0	0.100000E 01
279	4111EB85	16657	-5243	0.112000E 01
280	411428F5	16660	10485	0.126000E 01
281	4116147A	16662	5242	0.138000E 01
282	4118A3D7	16664	-23593	0.154000E 01
283	411B851E	16667	-31458	0.172000E 01
284	411EE147	16670	-7865	0.193000E 01

285	41240000	16676	0	0.225000E 01
286	41300000	16688	0	0.300000E 01
287	00000000	0	0	
288	423E8000	16958	-32768	0.625000E 02
289	42155558	16917	21848	0.213334E 02
290	41D20003	16850	3	0.131250E 02
291	41979411	16791	-27631	0.947365E 01
292	41745D40	16756	23872	0.727277E 01
293	41600000	16736	0	0.600000E 01
294	4152F66B	16722	-2453	0.518516E 01
295	41449245	16708	-28091	0.428571E 01
296	413EDB6E	16702	-9362	0.392857E 01
297	4137D061	16695	-12191	0.348837E 01
298	4133E453	16691	-7085	0.324324E 01
299	412D5F84	16685	24452	0.283582E 01
300	412A72F0	16682	29424	0.265306E 01
301	4127FFFE	16679	-2	0.250000E 01
302	41280000	16680	0	0.250000E 01
303	412A72F0	16682	29424	0.265306E 01
304	41307C1D	16688	31773	0.303030E 01
305	412E50D7	16686	20695	0.289474E 01
306	4137D061	16695	-12191	0.348837E 01
307	413EDB6E	16702	-9362	0.392857E 01
308	41449245	16708	-28091	0.428571E 01
309	4152F66B	16722	-2453	0.518516E 01
310	41600000	16736	0	0.600000E 01
311	41745D38	16756	23864	0.727276E 01
312	41979415	16791	-27627	0.947365E 01
313	41D1FFE6	16849	-26	0.131250E 02
314	42155559	16917	21849	0.213334E 02
315	423E7FF7	16958	32759	0.624999E 02
316	FFFF0001	-1	1	
317	FFE90000	-23	0	
318	00000000	0	0	
319	FFE70001	-25	1	
320	00020002	2	2	
321	00000000	0	0	
322	00000002	0	2	
323	00000003	0	3	
324	0000000A	0	10	
325	00000012	0	18	
326	FFE60003	-26	3	
327	FFE90000	-23	0	
328	00000000	0	0	
329	FFE70000	-25	0	
330	00020002	2	2	
331	00010000	1	0	
332	40C00000	16576	0	0.750000E 00
333	41100000	16656	0	0.100000E 01
334	00000002	0	2	
335	00000003	0	3	
336	000001F4	0	500	
337	00000152	0	338	
338	0000015D	0	349	
339	43103000	17168	12288	0.259000E 03
340	44704D00	17520	19712	0.287490E 05
341	43103000	17168	12288	0.259000E 03
342	44704D00	17520	19712	0.287490E 05
343	00000003	0	3	
344	00000001	0	1	
345	43103000	17168	12288	0.259000E 03
346	00000002	0	2	
347	00000000	0	0	
348	00000000	0	0	
349	00000003	0	3	
350	426F0000	17007	0	0.111000E 03
351	44302100	17456	8448	0.123210E 05
352	426F0000	17007	0	0.111000E 03
353	44302100	17456	8448	0.123210E 05
354	00000001	0	1	
355	00000001	0	1	
356	426F0000	17007	0	0.111000E 03

357	00000002	0	2
358	00000000	0	0
359	00000000	0	0
360	00000001	0	1
361	00000000	0	0
362	00000001	0	1
363	000000B9	0	185
364	00000000	0	0
365	00020000	2	0
366	01010000	257	0
367	00000000	0	0
368	00000006	0	6
369	00000000	0	0
370	00000000	0	0
371	00000000	0	0
372	00000000	0	0
373	00000000	0	0
374	00000000	0	0
375	00000000	0	0
376	00000000	0	0
377	00000000	0	0
378	00000000	0	0
379	00000000	0	0
380	00000000	0	0
381	00000190	0	400
382	00000000	0	0
383	000000B4	0	180
384	00000000	0	0
385	000E0000	14	0
386	01010000	257	0
387	00000000	0	0
388	00000002	0	2
389	00000000	0	0
390	00000000	0	0
391	00000000	0	0
392	00000000	0	0
393	00000000	0	0
394	00000000	0	0
395	00000000	0	0
396	00000000	0	0
397	00000000	0	0
398	00000000	0	0
399	00000000	0	0
400	00000000	0	0
401	000001A4	0	420
402	00000000	0	0
403	00000094	0	148
404	00000000	0	0
405	00000000	0	0
406	01010002	257	2
407	00000094	0	148
408	00000005	0	5
409	00000002	0	2
410	00000000	0	0
411	00000000	0	0
412	00000000	0	0
413	00000000	0	0
414	00000000	0	0
415	00000000	0	0
416	00000000	0	0
417	00000000	0	0
418	00000000	0	0
419	00000000	0	0
420	00000000	0	0
421	000001B8	0	440
422	00000000	0	0
423	00000000	0	0
424	00000000	0	0
425	00000000	0	0
426	00000000	0	0
427	00000000	0	0
428	00000000	0	0

429	00000000	0	0
430	00000000	0	0
431	00000000	0	0
432	00000000	0	0
433	00000000	0	0
434	00000000	0	0
435	00000000	0	0
436	00000000	0	0
437	00000000	0	0
438	00000000	0	0
439	00000000	0	0
440	00000000	0	0
441	00000100	0	460
442	00000000	0	0
443	00000000	0	0
444	00000000	0	0
445	00000000	0	0
446	00000000	0	0
447	00000000	0	0
448	00000000	0	0
449	00000000	0	0
450	00000000	0	0
451	00000000	0	0
452	00000000	0	0
453	00000000	0	0
454	00000000	0	0
455	00000000	0	0
456	00000000	0	0
457	00000000	0	0
458	00000000	0	0
459	00000000	0	0
460	00000000	0	0
461	000001E0	0	480
462	00000000	0	0
463	00000000	0	0
464	00000000	0	0
465	00000000	0	0
466	00000000	0	0
467	00000000	0	0
468	00000000	0	0
469	00000000	0	0
470	00000000	0	0
471	00000000	0	0
472	00000000	0	0
473	00000000	0	0
474	00000000	0	0
475	00000000	0	0
476	00000000	0	0
477	00000000	0	0
478	00000000	0	0
479	00000000	0	0
480	00000000	0	0
481	00000000	0	0
482	00000000	0	0
483	00000000	0	0
484	00000000	0	0
485	00000000	0	0
486	00000000	0	0
487	00000000	0	0
488	00000000	0	0
489	00000000	0	0
490	00000000	0	0
491	00000000	0	0
492	00000000	0	0
493	00000000	0	0
494	00000000	0	0
495	00000000	0	0
496	00000000	0	0
497	00000000	0	0
498	00000000	0	0
499	00000000	0	0
500	00000000	0	0

PROGRAM USED TO LOAD X-VECTCR AND CALL SIMULATION ROUTINE

```
      IMPLICIT INTEGER(A-Q,S-Z),REAL(R)                                 WIL00010
      INTEGER*4 RTERM                                                   WIL00020
      INTEGER*2 HW(2)                                                   WIL00030
      DIMENSION X(500),RX(500)                                          WIL00040
      EQUIVALENCE (FW1,HW(1)),(X,RX)                                    WIL00050
      LOGICAL TR6,TRACE,TRARG,TRBLO,TRUPD,TRSCN,TRULK,TRS1              WIL00060
      CCMMCN/XVECTR/ X                                                  WIL00070
      COMMON/NLPTMP/ OUT6,RTERM,WTERM,TR6                               WIL00080
      NAMELIST/P/ OUT6,TR6,RTERM,WTERM                                  WIL00090
      DO 8 II=1,500                                                     WIL00100
      X(II)=0                                                           WIL00110
8     CONTINUE                                                          WIL00120
      TR6=.FALSE.                                                       WIL00130
      OUT6=6                                                            WIL00140
      RTERM=5                                                           WIL00150
      WTERM=6                                                           WIL00160
      WRITE(6,4)                                                        WIL00170
4     FORMAT(' ENTER OPTIONAL DATA')                                    WIL00180
      READ(5,P)                                                         WIL00190
      READ(4,1) NRNG                                                    WIL00200
1     FORMAT(I5)                                                        WIL00210
      DO 5 I=1,NRNG                                                     WIL00220
      READ(4,6) LINE,ARTYP,R1,R2                                        WIL00230
      IF(LINE.NE.I) WRITE(6,2)                                          WIL00240
2     FORMAT('OSEQUENCE ERROR')                                         WIL00250
6     FORMAT(I3,I7,2F10.0)                                              WIL00260
      IF(ARTYP.NE.1) GOTO 9                                             WIL00270
      HW(1)=R1                                                          WIL00280
      HW(2)=R2                                                          WIL00290
      X(I)=FW1                                                          WIL00300
      GOTO 5                                                            WIL00310
9     IF(ARTYP.EQ.0) GOTO 10                                            WIL00320
      RX(I)=R2                                                          WIL00330
      GOTO 5                                                            WIL00340
10    X(I)=R2                                                           WIL00350
5     CCNTINUE                                                          WIL00360
      CALL SIMULT                                                       WIL00370
      WRITE(6,3)                                                        WIL00380
3     FORMAT(' SIMULATION TERMINATED')                                  WIL00390
      END                                                               WIL00400
```

APPENDIX B

DICTIONARY OF PROGRAM VARIABLES

ACLOCK	Absolute Clock Time.
ALTBLO	Alternate Block Number.
AVAIL	Amount of Storage currently available.
BASE	Value to which spread will be added in GENERATE and ADVANCE blocks.
CBLO	Pointer to current Block being processed.
COMVAL	Comparison value used in a SELECT Block.
CQUE	Pointer to current Queue being processed.
CREATE	Time from current clock time until Transaction due to leave FEC.
GSTO	Pointer to current Storage being processed.
CTRA	Pointer to current Transaction being processed.
DELAY	Time delay caused by ADVANCE Block.
DTIME	Time differential between clock and last entry into a Queue or Storage.
FOS( )	Stack for evaluating floating point arguments.
FTRA	Pointer to first Transaction on list of unused Transactions.
FREWDT	Width of frequency interval in a Table entity.
GIVEUP	Number of Storage units being made available.
INST	Pointer to current block element being processed.

INTDEC( )	Indicates whether corresponding OS/FOS value is integer or floating point.
HIGH	Ending entity number to be used in SELECT Block search.
JBL	Number of Block whose routine is being executed.
KDEX	Index used to keep count of number of arguments specified for a Block.
KMODE	Special variable used to determine a unique routine to be performed for a given set of conditions.
LOW	Beginning entity number to be used in SELECT Block search.
MAXCTS	Maximum Contents of Storage or Queue.
MAXX	Maximum number of X-Vector elements.
MDFR	Modifier used in TEST, SELECT and GATE blocks.
MODE	Indicates whether simulation will begin in Clear, Reset or Restart mode.
MRNG	Number of points in a Function.
NEWBDT	Block departure time of Transaction being merged into FEC.
NEWPRI	Priority of Transaction being merged into CEC.
NEXTBDT	Block Departure Time of first Transaction on FEC.
NEXBLO	Next Block number Transaction is to enter.
NOUPD	Indicator used to merge a Transaction into CEC without updating Clock.
OLDBDT	Block Departure Time of Transactions being checked in FEC.
OLDPRI	Priority of Transactions being checked in CEC.

OPCODE	Operation Code indicating type of Block.
OS( )	Stack for evaluating integer arguments.
PCONTS	Present contents of Queue.
PNTA	Pointer to last entry on return address stack.
PNTB	Pointer to last evaluated argument on OS/FOS stack.
PNTF	Pointer to first element of frequency intervals for a Table entity.
PNTS	Pointer to first element of Slope values of current Function entity.
PNTT	Pointer to Transaction being compared with current Transaction.
PNTX	Pointer to first element of independent Variable values of current Function entity.
PNTY	Pointer to first element of Function values of current Function entity.
RAS	Return Address Stack used to temporarily hold location of next arguments to be evaluated.
SCFLAG	Status Change Flag. Set equal to 1 when certain entity attributes change and 0 otherwise.
SE	Pointer to top of push-down chain for empty Storage condition.
SF	Pointer to top of push-down chain for full Storage condition.
SNE	Pointer to top of push-down chain for not empty Storage condition.
SNF	Pointer to top of Push-down chain for not full Storage Condition.
SRED	Pointer to top of Push-down chain for reduced contents Storage condition.

SSIND	Scan Status Indicator. True if Transaction active in CEC. False if Transaction inactive in CEC.
TASGN	The value being assigned in an ASSIGN block.
TBLNO	Number of Table being tabulated into.
TEST	Value of pointers associated with delay chains.
TPAR	Parameter being assigned value in an ASSIGN Block.
TRCNT	Indicates the Transaction number being assigned a new Transaction.
TRSIZE	Indicates size of X-Vector printout.
TYPARG	Indicates whether search argument of a Function entity is integer or floating point.
TYPX	Indicates whether independent variable of Function entity is integer or floating point.
TYPY	Indicates whether function values of Function entity are integer or floating point.
ULLI	Indicates upper limit of lower interval for a Table entity.
UNITS	Number of units entering or leaving a Queue.
USED	Current contents of Storage.
WANT	Number of Storage units required.
WTFAC	Weighting Factor for use in Tabulate block. Normally equal to 1.
ZRTN	Holds statement number for use in returning from various routines.

```
************************  SIMULATION ROUTINE  ****************************

      SUBROUTINE SIMULT                                                 WIL00410
      IMPLICIT INTEGER(A-Q,S-Z),REAL(R)                                 WIL00420
      INTEGER*4 X(500),FWORD(4),RAS(10),OS(20),SEED(8),RTERM            WIL00430
      INTEGER*2 HWORD(8),CTRA,FTRA,ZRTN,PNTA,PNTE,PNTT,INTDEC(20),PNTF, WIL00440
     -INST,PNTX,PNTY,PNTS,WRTN,TYPX,TYPY,CBLO,NEXBLO,MDFR,              WIL00450
     -HW1,HW2,HW3,HW4,HW5,HW6,HW7,HW8                                   WIL00460
      REAL*8 DWORD(2),DW1,DW2                                           WIL00470
      REAL*4 FFWORD(4),FOS(20),RX(500),FFW1,FFW2,FFW3,FFW4              WIL00480
      LOGICAL*1 BYTE(16),SSIND                                          WIL00490
      LOGICAL TR6,TRACE,TRARG,TRBLO,TRUPD,TRSCN,TRULK,TRS1              WIL00500
      COMMON/XVECTR/ X                                                  WIL00510
      COMMON/NLPTMP/ OUT6,RTERM,WTERM,TR6                               WIL00520
      EQUIVALENCE(DWORD,FWORD,FFWORD,HWORD,BYTE),(DWORD(1),DW1),        WIL00530
     -(DWORD(2),DW2),                                                   WIL00540
     -(FWORD(1),FW1),(FWORD(2),FW2),(FWORD(3),FW3),(FWORD(4),FW4),      WIL00550
     -(FFWORD(1),FFW1),(FFWORD(2),FFW2),(FFWORD(3),FFW3),               WIL00560
     -(FFWORD(4),FFW4),                                                 WIL00570
     -(HWORD(1),HW1),(HWORD(2),HW2),(HWORD(3),HW3),(HWORD(4),HW4),      WIL00580
     -(HWORD(5),HW5),(HWORD(6),HW6),(HWORD(7),HW7),(HWORD(8),HW8),      WIL00590
     -(X(1),MODE),(X(2),TRMCNT),(X(3),SEED),(X(11),CLOCK),(X(12),NSTO), WIL00600
     -(X(13),STO),(X(14),NQUE),(X(15),QUE),(X(16),NFUNCT),(X(17),FUNCT),WIL00610
     -(X(18),NBLO),(X(19),BLOCK),(X(20),NTAB),(X(21),TAB),(X(22),NSAV), WIL00620
     -(X(23),SAVE),(X(24),NPAR),(X(25),TRAN),(X(26),FTCEC),             WIL00630
     -(X(27),LTCEC),(X(28),FTFEC),(X(29),LTFEC),(X(30),ERR),            WIL00640
     -(X(31),NVAR),(X(32),VAR),                                         WIL00650
     -(X,RX),(OS,FOS)                                                   WIL00660
      NAMELIST/P/ TR6,TRACE,TRARG,TRBLO,TRUPD,TRSCN,TRULK,OUT6,OUTX,    WIL00670
     -TRSIZE,TRS1                                                       WIL00680
C                                   SIMULATE                            WIL00690
      TRACE=.TRUE.                                                      WIL00700
      TRARG=.TRUE.                                                      WIL00710
      TRBLO=.TRUE.                                                      WIL00720
      TRUPD=.TRUE.                                                      WIL00730
      TRSCN=.TRUE.                                                      WIL00740
      TRULK=.TRUE.                                                      WIL00750
      TRS1=.FALSE.                                                      WIL00760
      OUTX=0                                                            WIL00770
      MAXX=500                                                          WIL00780
      TRSIZE=MAXX                                                       WIL00790
      IF(TR6) WRITE(WTERM,9025)                                         WIL00800
      IF(TR6) READ(RTERM,P)                                             WIL00810
      IF(MODE.NE.4) GOTO 15                                             WIL00820
      ZRTN=4005                                                         WIL00830
      GOTO 2311                                                         WIL00840
```

```
 15   IF(MODE.EQ.3) GOTO 50                                          WIL00850
      IF(MODE.EQ.2) GOTO 60                                          WIL00860
      IF(TRACE) WRITE(OUT6,9003)                                     WIL00870
C                                          SLOPE COMPUTATION        WIL00880
      IF(TRACE) WRITE(OUT6,9002)                                     WIL00890
      K=0                                                            WIL00900
 20   IF(K.GE.NFUNCT) GOTO 25                                        WIL00910
      K=K+1                                                          WIL00920
      PNTF=X(FUNCT+K)                                                WIL00930
      FW1=X(PNTF+5)                                                  WIL00940
      IF(HW1.GE.2) GOTO 25                                           WIL00950
      MRNG=HW2                                                       WIL00960
      PNTX=PNTF+6                                                    WIL00970
      PNTY=PNTX+MRNG                                                 WIL00980
      PNTS=PNTY+MRNG                                                 WIL00990
      FW1=X(PNTF+6)                                                  WIL01000
      TYPX=HW1                                                       WIL01010
      TYPY=HW2                                                       WIL01020
      RX(PNTS+1)=0.0                                                 WIL01030
      DO 22 I=2,MRNG                                                 WIL01040
      KXVAL=X(PNTX+I)-X(PNTX+I-1)                                    WIL01050
      RYVAL=X(PNTY+I)-X(PNTY+I-1)                                    WIL01060
      IF(TYPY.EQ.1) RYVAL=RX(PNTY+I)-RX(PNTY+I-1)                    WIL01070
      IF(TYPX.EQ.1) GOTO 21                                          WIL01080
      RX(PNTS+I)=RYVAL/KXVAL                                         WIL01090
      GOTO 22                                                        WIL01100
 21   RXVAL=RX(PNTX+I)-RX(PNTX+I-1)                                  WIL01110
      RX(PNTS+I)=RYVAL/RXVAL                                         WIL01120
 22   CONTINUE                                                       WIL01130
      GOTO 20                                                        WIL01140
C                                          BEGIN NEW SIMULATION     WIL01150
 25   ACLOCK=0                                                       WIL01160
 26   FTRA=TRAN                                                      WIL01170
C                                          CREATE CHAIN OF UNUSED   WIL01180
C                                          TRANSACTIONS USING REMAINING WIL01190
C                                          X VECTOR SPACE           WIL01200
      KK=MAXX-TRAN                                                   WIL01210
      KK=KK/(8+NPAR)-1                                               WIL01220
      LL=TRAN                                                        WIL01230
      DO 27 I=1,KK                                                   WIL01240
      HW1=0                                                          WIL01250
      HW2=LL+8+NPAR                                                  WIL01260
      X(LL+1)=FW1                                                    WIL01270
      LL=LL+8+NPAR                                                   WIL01280
 27   CONTINUE                                                       WIL01290
C                                          SEARCH FOR GENERATE BLOCKS AND WIL01300
C                                          PLACE ONE TRANSACTION ON FEC FORWIL01310
C                                          EACH ONE FOUND           WIL01320
```

```
      TRCNT=0                                                           WIL01330
      JBL=0                                                             WIL01340
30    IF(JBL.EQ.NBLO) GOTO 300                                          WIL01350
      JBL=JBL+1                                                         WIL01360
      CBLO=X(BLOCK+JBL)                                                 WIL01370
      FW1=X(CBLO+1)                                                     WIL01380
      OPCODE=HW1                                                        WIL01390
      IF(TRACE) WRITE(OUT6,9038) OPCODE                                 WIL01400
      IF(OPCODE.NE.1) GOTO 30                                           WIL01410
31    INST=CBLO+2                                                       WIL01420
      ZRTN=32                                                           WIL01430
      GO TO 100                                                         WIL01440
32    ZRTN=30                                                           WIL01450
      GOTO 200                                                          WIL01460
C                                                   CLEAR               WIL01470
50    ACLOCK=0                                                          WIL01480
      IF(TRACE) WRITE(OUT6,9004)                                        WIL01490
      CLOCK=0                                                           WIL01500
      FTCEC=0                                                           WIL01510
      LTCEC=0                                                           WIL01520
      FTFEC=0                                                           WIL01530
      LTFEC=0                                                           WIL01540
      FTRA=TRAN                                                         WIL01550
      KK=MAXX-TRAN                                                      WIL01560
      KK=KK/(8+NPAR)-1                                                  WIL01570
      LL=TRAN                                                           WIL01580
      DO 40 I=1,KK                                                      WIL01590
      HW1=0                                                             WIL01600
      HW2=LL+8+NPAR                                                     WIL01610
      X(LL+1)=FW1                                                       WIL01620
      LL=LL+8+NPAR                                                      WIL01630
40    CONTINUE                                                          WIL01640
      IF(NSTO.EQ.0) GOTO 52                                             WIL01650
      DO 51 I=1,NSTO                                                    WIL01660
      N=X(STO+I)                                                        WIL01670
      X(N+1)=0                                                          WIL01680
      X(N+2)=X(N+2)+X(N+1)                                              WIL01690
      DW1=1.0                                                           WIL01700
      X(N+3)=FW1                                                        WIL01710
      X(N+4)=FW2                                                        WIL01720
      X(N+5)=0                                                          WIL01730
      X(N+6)=0                                                          WIL01740
      X(N+7)=0                                                          WIL01750
51    CONTINUE                                                          WIL01760
52    IF(NQUE.EQ.0) GOTO 54                                             WIL01770
      DO 53 I=1,NQUE                                                    WIL01780
      N=X(QUE+I)                                                        WIL01790
      X(N+1)=0                                                          WIL01800
```

```
      X(N+2)=0                                                          WIL01810
      X(N+3)=0                                                          WIL01820
      DW1=1.0                                                           WIL01830
      X(N+4)=FW1                                                        WIL01840
      X(N+5)=FW2                                                        WIL01850
      X(N+6)=0                                                          WIL01860
      X(N+7)=0                                                          WIL01870
53    CONTINUE                                                          WIL01880
54    IF(NSAV.EQ.0) GOTO 56                                             WIL01890
      DO 55 I=1,NSAV                                                    WIL01900
      X(SAVE+I)=0                                                       WIL01910
55    CONTINUE                                                          WIL01920
56    IF(NTAB.EQ.0) GOTO 58                                             WIL01930
      DO 57 I=1,NTAB                                                    WIL01940
      N=X(TAB+I)                                                        WIL01950
      RX(N+1)=0.0                                                       WIL01960
      RX(N+2)=0.0                                                       WIL01970
      RX(N+3)=0.0                                                       WIL01980
      RX(N+4)=0.0                                                       WIL01990
      RX(N+6)=0.0                                                       WIL02000
      K=X(N+9)                                                          WIL02010
      DO 57 J=1,K                                                       WIL02020
      X(N+10+J)=0                                                       WIL02030
57    CONTINUE                                                          WIL02040
58    GOTO(59,300,26),MODE                                              WIL02050
59    CALL ERROR(603,1,&9999)                                           WIL02060
C                                                RESET                  WIL02070
60    ACLOCK=ACLOCK+CLOCK                                               WIL02080
      IF(TRACE) WRITE(OUT6,9005)                                        WIL02090
      CLOCK=0                                                           WIL02100
      IF(NSTO.EQ.0) GOTO 62                                             WIL02110
      DO 61 I=1,NSTO                                                    WIL02120
      N=X(STO+I)                                                        WIL02130
      DW1=1.0                                                           WIL02140
      X(N+3)=FW1                                                        WIL02150
      X(N+4)=FW2                                                        WIL02160
      X(N+5)=ACLOCK                                                     WIL02170
      X(N+6)=X(N+1)                                                     WIL02180
      X(N+7)=X(N+1)                                                     WIL02190
61    CONTINUE                                                          WIL02200
62    IF(NQUE.EQ.0) GOTO 56                                             WIL02210
      DO 63 I=1,NQUE                                                    WIL02220
      N=X(QUE+I)                                                        WIL02230
      X(N+1)=ACLOCK                                                     WIL02240
      DW1=1.0                                                           WIL02250
      X(N+4)=FW1                                                        WIL02260
      X(N+5)=FW2                                                        WIL02270
      X(N+2)=X(N+6)                                                     WIL02280
```

```
      X(N+3)=0                                                          WIL02290
      X(N+7)=X(N+6)                                                     WIL02300
 63   CONTINUE                                                          WIL02310
      GOTO 56                                                           WIL02320
C                                                ARGUMENT EVALUATION    WIL02330
 100  FNTA=0                                                            WIL02340
      FNTB=0                                                            WIL02350
      IF(TRACE) WRITE(OUT6,9006)                                        WIL02360
 1001 FW1=X(INST)                                                       WIL02370
      IF(HW1.GE.0) GOTO 190                                             WIL02380
      IF(HW2) 1002,1004,1003                                            WIL02390
 1002 HW2=-HW2                                                          WIL02400
      IF(HW2.GT.NPAR) CALL ERROR(492,1,&9999)                           WIL02410
      IF(HW2.LE.0) CALL ERROR(492,2,&9999)                              WIL02420
      HW2=X(CTRA+8+HW2)                                                 WIL02430
 1003 HW1=-HW1                                                          WIL02440
      GOTO (101,102,103,104,105,106,107,108,109,                        WIL02450
     -110,111,112,113,114,115,116,117,118,119,                           WIL02460
     -120,121,122,123,124,125,126,127,128,129,                           WIL02470
     -130,131,132,133),HW1                                              WIL02480
 1004 IF(HW1.EQ.-33) GOTO 133                                           WIL02490
      IF(HW1.EQ.-23) GOTO 123                                           WIL02500
      IF(HW1.EQ.-25) GOTO 125                                           WIL02510
      CALL ERROR(45,0,&9999)                                            WIL02520
C                                                P                      WIL02530
 101  FW1=X(CTRA+8+HW2)                                                 WIL02540
      GO TO 190                                                         WIL02550
C                                                PR                     WIL02560
 102  FW1=X(CTRA+5)                                                     WIL02570
      HW1=0                                                             WIL02580
      GOTO 190                                                          WIL02590
C                                                M1                     WIL02600
 103  FW1=CLOCK-X(CTRA+4)                                               WIL02610
      GOTO 190                                                          WIL02620
C                                                MP                     WIL02630
 104  FW1=CLOCK-X(CTRA+8+HW2)                                           WIL02640
      GOTO 190                                                          WIL02650
C                                                SC                     WIL02660
 105  IF(HW2.GT.NSTO) CALL ERROR(499,0,&9999)                           WIL02670
      N=X(STO+HW2)                                                      WIL02680
      FW1=X(N+6)                                                        WIL02690
      GOTO 190                                                          WIL02700
C                                                SM                     WIL02710
 106  IF(HW2.GT.NSTO) CALL ERROR(499,0,&9999)                           WIL02720
      N=X(STO+HW2)                                                      WIL02730
      FW1=X(N+7)                                                        WIL02740
      GOTO 190                                                          WIL02750
C                                                SR                     WIL02760
```

```
 107  IF(HW2.GT.NSTO) CALL ERROR(499,0,&9999)                    WIL02770
      N=X(STO+HW2)                                               WIL02780
      FW3=X(N+3)                                                 WIL02790
      FW4=X(N+4)                                                 WIL02800
      FW1=X(N+1)+X(N+2)                                          WIL02810
      FW1=FW1*CLOCK                                              WIL02820
      FW1=(DW2/FW1)*1000                                         WIL02830
      GOTO 190                                                   WIL02840
C                                                          SA    WIL02850
 108  IF(HW2.GT.NSTO) CALL ERROR(499,0,&9999)                    WIL02860
      N=X(STO+HW2)                                               WIL02870
      FW3=X(N+3)                                                 WIL02880
      FW4=X(N+4)                                                 WIL02890
      FW1=DW2/CLOCK                                              WIL02900
      GOTO 190                                                   WIL02910
C                                                           S    WIL02920
 109  IF(HW2.GT.NSTO) CALL ERROR(499,0,&9999)                    WIL02930
      N=X(STO+HW2)                                               WIL02940
      FW1=X(N+1)                                                 WIL02950
      GOTO 190                                                   WIL02960
C                                                           R    WIL02970
 110  IF(HW2.GT.NSTO) CALL ERROR(499,0,&9999)                    WIL02980
      N=X(STO+HW2)                                               WIL02990
      FW1=X(N+2)                                                 WIL03000
      GOTO 190                                                   WIL03010
C                                                          ST    WIL03020
 111  IF(HW2.GT.NSTO) CALL ERROR(499,0,&9999)                    WIL03030
      N=X(STO+HW2)                                               WIL03040
      FW3=X(N+3)                                                 WIL03050
      FW4=X(N+4)                                                 WIL03060
      FW1=DW2/X(N+6)                                             WIL03070
      GOTO 190                                                   WIL03080
C                                                           Q    WIL03090
 112  IF(HW2.GT.NQUE) CALL ERROR(500,0,&9999)                    WIL03100
      N=X(QUE+HW2)                                               WIL03110
      FW1=X(N+6)                                                 WIL03120
      GOTO 190                                                   WIL03130
C                                                          QA    WIL03140
 113  IF(HW2.GT.NQUE) CALL ERROR(500,0,&9999)                    WIL03150
      N=X(QUE+HW2)                                               WIL03160
      FW3=X(N+4)                                                 WIL03170
      FW4=X(N+5)                                                 WIL03180
      FW1=DW2/CLOCK                                              WIL03190
      GOTO 190                                                   WIL03200
C                                                          QM    WIL03210
 114  IF(HW2.GT.NQUE) CALL ERROR(500,0,&9999)                    WIL03220
      N=X(QUE+HW2)                                               WIL03230
      FW1=X(N+7)                                                 WIL03240
```

```
      GOTO 190                                                          WIL03250
C                                                   QC                  WIL03260
  115 IF(HW2.GT.NQUE) CALL ERROR(500,0,&9999)                           WIL03270
      N=X(QUE+HW2)                                                      WIL03280
      FW1=X(N+2)                                                        WIL03290
      GOTO 190                                                          WIL03300
C                                                   QZ                  WIL03310
  116 IF(HW2.GT.NQUE) CALL ERROR(500,0,&9999)                           WIL03320
      N=X(QUE+HW2)                                                      WIL03330
      FW1=X(N+3)                                                        WIL03340
      GOTO 190                                                          WIL03350
C                                                   QT                  WIL03360
  117 IF(HW2.GT.NQUE) CALL ERROR(500,0,&9999)                           WIL03370
      N=X(QUE+HW2)                                                      WIL03380
      FW3=X(N+4)                                                        WIL03390
      FW4=X(N+5)                                                        WIL03400
      FW1=DW2/X(N+2)                                                    WIL03410
      GOTO 190                                                          WIL03420
C                                                   QX                  WIL03430
  118 IF(HW2.GT.NQUE) CALL ERROR(500,0,&9999)                           WIL03440
      N=X(QUE+HW2)                                                      WIL03450
      FW3=X(N+4)                                                        WIL03460
      FW4=X(N+5)                                                        WIL03470
      FW1=X(N+2)-X(N+3)                                                 WIL03480
      FW1=DW2/FW1                                                       WIL03490
      GOTO 190                                                          WIL03500
C                                                   TB                  WIL03510
  119 IF(HW2.GT.NTAB) CALL ERROR(435,0,&9999)                           WIL03520
      N=X(TAB+HW2)                                                      WIL03530
      FFW1=RX(N+1)/X(N+6)                                               WIL03540
      GOTO 190                                                          WIL03550
C                                                   TC                  WIL03560
  120 IF(HW2.GT.NTAB) CALL ERROR(435,0,&9999)                           WIL03570
      N=X(TAB+HW2)                                                      WIL03580
      FW1=X(N+6)                                                        WIL03590
      GOTO 190                                                          WIL03600
C                                                   TD                  WIL03610
  121 IF(HW2.GT.NTAB) CALL ERROR(435,0,&9999)                           WIL03620
      N=X(TAB+HW2)                                                      WIL03630
      FFW1=RX(N+2)-(RX(N+1)**2)/X(N+6)                                  WIL03640
      FFW1=SQRT(FFW1/(X(N+6)-1))                                        WIL03650
      GOTO 190                                                          WIL03660
C                                                   FN                  WIL03670
  122 IF(HW2.GT.NFUNCT) CALL ERROR(507,0,&9999)                         WIL03680
      PNTA=PNTA+1                                                       WIL03690
      RAS(PNTA)=INST+1                                                  WIL03700
      INST=X(FUNCT+HW2)+1                                               WIL03710
      GOTO 1001                                                         WIL03720
```

```
123   FW1=X(INST+3)                                                  WIL03730
      MRNG=HW2                                                       WIL03740
      PNTX=INST+4                                                    WIL03750
      PNTY=PNTX                                                      WIL03760
      IF(HW1.LE.3) PNTY=PNTY+MRNG                                    WIL03770
      PNTS=PNTY+MRNG                                                 WIL03780
      KARG=OS(PNTB)                                                  WIL03790
      RKARG=FOS(PNTB)                                                WIL03800
      TYPARG=INTDEC(PNTB)                                            WIL03810
      KMODE=2*(HW1-1)+TYPARG+1                                       WIL03820
      FW2=X(INST+4)                                                  WIL03830
      TYPX=HW3                                                       WIL03840
      TYPY=HW4                                                       WIL03850
      IF(KMODE.GT.7) GOTO 1235                                       WIL03860
      KCUNT=0                                                        WIL03870
      IF(TYPARG.EQ.1.AND.TYPX.EQ.0) KARG=RKARG                       WIL03880
      IF(TYPARG.EQ.0.AND.TYPX.EQ.0) GOTO 1232                        WIL03890
      IF(TYPARG.EQ.0.AND.TYPX.EQ.1) RKARG=KARG                       WIL03900
      DO 1231 I=1,MRNG                                               WIL03910
      KCUNT=KOUNT+1                                                  WIL03920
      IF(RKARG.LE.RX(PNTX+KCUNT)) GO TO 1234                         WIL03930
1231  CONTINUE                                                       WIL03940
      GOTO 1234                                                      WIL03950
1232  DO 1233 I=1,MRNG                                               WIL03960
      KCUNT=KOUNT+1                                                  WIL03970
      IF(KARG.LE.X(PNTX+KOUNT)) GOTO 1234                            WIL03980
1233  CONTINUE                                                       WIL03990
1234  IF(KMODE.LE.2) GOTO 1237                                       WIL04000
      KARG=KOUNT                                                     WIL04010
      TYPARG=0                                                       WIL04020
      KMODE=KMODE+4                                                  WIL04030
1235  IF(TYPARG.EQ.1) KARG=RKARG                                     WIL04040
      IF(KARG.GT.MRNG) CALL ERROR(509,0,&9999)                       WIL04050
      IF(KARG.LT.1) CALL ERROR(509,0,&9999)                          WIL04060
      IF(KMODE.GE.9) GOTO 1236                                       WIL04070
      OS(PNTB)=X(PNTY+KARG)                                          WIL04080
      INTDEC(PNTB)=TYPY                                              WIL04090
      GOTO 125                                                       WIL04100
1236  X(INST+1)=X(PNTY+KARG)                                         WIL04110
      PNTB=PNTB-1                                                    WIL04120
      INST=INST+1                                                    WIL04130
      GOTO 1001                                                      WIL04140
1237  KXVAL=X(PNTX+KOUNT-1)                                          WIL04150
      RXVAL=RX(PNTX+KOUNT-1)                                         WIL04160
      KYVAL=X(PNTY+KOUNT-1)                                          WIL04170
      RYVAL=RX(PNTY+KOUNT-1)                                         WIL04180
      RSLOPE=RX(PNTS+KOUNT)                                          WIL04190
      INTDEC(PNTB)=1                                                 WIL04200
```

```
      IF(KOUNT.EQ.1) KOUNT=2                                            WIL04210
      IF(KMODE.EQ.1) GOTO 1238                                          WIL04220
      FFW1=RX(PNTX+MRNG)                                                WIL04230
      IF(RKARG.GT.FFW1) RKARG=FFW1                                      WIL04240
      RXVAL=RKARG-RXVAL                                                 WIL04250
      RXVAL=RSLOPE*RXVAL                                                WIL04260
      FOS(PNTB)=RYVAL+RXVAL                                             WIL04270
      IF(TYPY.EQ.0) FOS(PNTB)=KYVAL+RXVAL                               WIL04280
      GOTO 125                                                          WIL04290
 1238 FW1=X(PNTX+MRNG)                                                  WIL04300
      IF(KARG.GT.FW1) KARG=FW1                                          WIL04310
      KXVAL=KARG-KXVAL                                                  WIL04320
      RXVAL=RSLOPE*KXVAL                                                WIL04330
      FOS(PNTB)=RYVAL+RXVAL                                             WIL04340
      IF(TYPY.EQ.0) FOS(PNTB)=KYVAL+RXVAL                               WIL04350
      GOTO 125                                                          WIL04360
C                                                   V                   WIL04370
 124  IF(HW2.GT.NVAR) CALL ERROR(514,0,&9999)                           WIL04380
      PNTA=PNTA+1                                                       WIL04390
      RAS(PNTA)=INST+1                                                  WIL04400
      INST=X(VAR+HW2)+1                                                 WIL04410
      GOTO 1001                                                         WIL04420
 125  INST=RAS(PNTA)                                                    WIL04430
      PNTA=PNTA-1                                                       WIL04440
      GOTO 1001                                                         WIL04450
C                                                   RN                  WIL04460
 126  IF(HW2.GT.8) CALL ERROR(507,0,&9999)                              WIL04470
      SEED(HW2)=SEED(HW2)*65539                                         WIL04480
      RN=0.5+SEED(HW2)*0.2328306E-9                                     WIL04490
      IF(PNTA.EQ.0) GOTO 1261                                           WIL04500
      FW1=X(RAS(PNTA)-1)                                                WIL04510
      IF(HW1.NE.-22) GOTO 1261                                          WIL04520
      PNTB=PNTB+1                                                       WIL04530
      FOS(PNTB)=RN                                                      WIL04540
      INTDEC(PNTB)=1                                                    WIL04550
      GOTO 191                                                          WIL04560
 1261 FW1=1000.*RN                                                      WIL04570
      GOTO 190                                                          WIL04580
C                                                   C1                  WIL04590
 127  FW1=CLOCK                                                         WIL04600
      GOTO 190                                                          WIL04610
C                                                   DUMMY               WIL04620
 128  FW1=X(SAVE+HW2)                                                   WIL04630
      GOTO 190                                                          WIL04640
 129  CALL ERROR(603,3,&9999)                                           WIL04650
C                                                   ARITHMETIC          WIL04660
 130  PNTB=PNTB-1                                                       WIL04670
      K=4*(HW2-1)+2*INTDEC(PNTB)+INTDEC(PNTB+1)+1                       WIL04680
```

```
      GOTO (1301,1302,1303,1304,1305,1306,1307,1308,1309,1310,1311,      WIL04690
     -1312,1313,1314,1315,1316,1317,1318,1319,1320),K                    WIL04700
 1301 OS(PNTB)=OS(PNTB)+OS(PNTB+1)                                       WIL04710
      GOTO 1321                                                          WIL04720
 1302 FOS(PNTB)=OS(PNTB)+FOS(PNTB+1)                                     WIL04730
      GOTO 1322                                                          WIL04740
 1303 FOS(PNTB)=FOS(PNTB)+OS(PNTB+1)                                     WIL04750
      GOTO 1322                                                          WIL04760
 1304 FOS(PNTB)=FOS(PNTB)+FOS(PNTB+1)                                    WIL04770
      GOTO 1322                                                          WIL04780
 1305 OS(PNTB)=OS(PNTB)-OS(PNTB+1)                                       WIL04790
      GOTO 1321                                                          WIL04800
 1306 FOS(PNTB)=OS(PNTB)-FOS(PNTB+1)                                     WIL04810
      GOTO 1322                                                          WIL04820
 1307 FOS(PNTB)=FOS(PNTB)-OS(PNTB+1)                                     WIL04830
      GOTO  1322                                                         WIL04840
 1308 FOS(PNTB)=FOS(PNTB)-FOS(PNTB+1)                                    WIL04850
      GOTO 1322                                                          WIL04860
 1309 OS(PNTB)=OS(PNTB)*OS(PNTB+1)                                       WIL04870
      GO TO 1321                                                         WIL04880
 131  CALL ERROR(603,4,&9999)                                            WIL04890
 1310 FOS(PNTB)=OS(PNTB)*FOS(PNTB+1)                                     WIL04900
      GOTO 1322                                                          WIL04910
 1311 FOS(PNTB)=FOS(PNTB)*OS(PNTB+1)                                     WIL04920
      GOTO 1322                                                          WIL04930
 1312 FOS(PNTB)=FOS(PNTB)*FOS(PNTB+1)                                    WIL04940
      GOTO 1322                                                          WIL04950
 1313 OS(PNTB)=OS(PNTB)/OS(PNTB+1)                                       WIL04960
      GOTO 1321                                                          WIL04970
 1314 FOS(PNTB)=OS(PNTB)/FOS(PNTB+1)                                     WIL04980
      GOTO 1322                                                          WIL04990
 1315 FOS(PNTB)=FOS(PNTB)/OS(PNTB+1)                                     WIL05000
      GOTO 1322                                                          WIL05010
 1316 FOS(PNTB)=FOS(PNTB)/FOS(PNTB+1)                                    WIL05020
      GOTO 1322                                                          WIL05030
 1317 OS(PNTB)=MOD(OS(PNTB),OS(PNTB+1))                                  WIL05040
      GOTO 1321                                                          WIL05050
 1318 FOS(PNTB)=AMOD(FLOAT(OS(PNTB)),FOS(PNTB+1))                        WIL05060
      GOTO 1322                                                          WIL05070
 1319 FOS(PNTB)=AMOD(FOS(PNTB),FLOAT(OS(PNTB+1)))                        WIL05080
      GOTO 1322                                                          WIL05090
 132  CALL ERROR(603,5,&9999)                                            WIL05100
 1320 FOS(PNTB)=AMOD(FOS(PNTB),FOS(PNTB+1))                              WIL05110
      GOTO 1322                                                          WIL05120
 1321 INTDEC(PNTB)=0                                                     WIL05130
      GOTO 191                                                           WIL05140
 1322 INTDEC(PNTB)=1                                                     WIL05150
      GOTO 191                                                           WIL05160
```

```
C                                                  RETURN FRCM ARG. EVALUATION
 133  MDFR=HW2
      I=1
      KDEX=PNTB
 1331 IF(I.GT.PNTB) GOTO 1333
      IF(INTDEC(PNTB).EQ.0) GOTO 1332
      IF(I.EQ.2.AND.(OPCODE.EQ.1.OR.OPCODE.EQ.2)) GOTO 1332
      OS(I)=FCS(I)
 1332 I=I+1
      GOTO 1331
 1333 IF(PNTB.EQ.7) GOTO 1334
      PNTB=PNTB+1
      OS(PNTB)=0
      INTDEC(PNTB)=0
      GOTO 1333
 1334 IF(.NOT.TRACE) GOTO 1337
      INTEGER*4 PR(7)
      IMAX=1
      DO 1335 I=1,7
      PR(I)=0
      IF(INTDEC(I).EQ.0) PR(I)=OS(I)
      IF(PR(I).NE.0) IMAX=I
 1335 CONTINUE
      IF(TRACE) WRITE(OUT6,9015) CBLO,OPCODE,MDFR,(PR(I),I=1,IMAX)
      DO 1336 I=1,7
      IF(INTDEC(I).EQ.1) WRITE(OUT6,9016) I,FOS(I)
 1336 CONTINUE
 1337 IF(ZRTN.EQ.32) GOTO 32
      IF(ZRTN.EQ.2316) GOTO 2316
      IF(ZRTN.EQ.4004) GOTO 4004
      CALL ERROR(603,6,&9999)
C                                                  PLACE EVALUATED ARG. ON STACK
 190  PNTB=PNTB+1
      OS(PNTB)=FW1
      INTDEC(PNTB)=0
 191  INST=INST+1
      GOTO 1001
C                                                  BLOCK ROUTINES
 200  IF(TRACE) WRITE(OUT6,9007) OPCODE,CBLO
      GOTO (201,202,203,204,205,206,207,208,209,210,211,212,213,214,
     -215,216,217,218,219,220,229,229,229,229,229,229,229,229,229,
     -230),OPCODE
C                                                  GENERATE
 201  IF(KDEX.GE.4.AND.OS(4).LE.TRCNT) GOTO 2014
      CTRA=FTRA
      FW1=X(CTRA+1)
      FTRA=HW2
      K=8+NPAR
```

```
      DO 2015 I=1,K                                                     WIL05650
      X(CTRA+I)=0                                                       WIL05660
 2015 CONTINUE                                                          WIL05670
      IF(HW2.EQ.0) CALL ERROR(468,0,&9999)                              WIL05680
      IF(TRBLO) WRITE(OUTX,9036) FTRA                                   WIL05690
      HW1=0                                                             WIL05700
      HW2=JBL                                                           WIL05710
      X(CTRA+1)=FW1                                                     WIL05720
      TRCNT=TRCNT+1                                                     WIL05730
      HW1=0                                                             WIL05740
      HW2=TRCNT                                                         WIL05750
      X(CTRA+8)=FW1                                                     WIL05760
      HW1=JBL+1                                                         WIL05770
      HW2=OS(5)                                                         WIL05780
      X(CTRA+5)=FW1                                                     WIL05790
      BYTE(1)=.TRUE.                                                    WIL05800
      BYTE(2)=.TRUE.                                                    WIL05810
      HW2=0                                                             WIL05820
      X(CTRA+6)=FW1                                                     WIL05830
      FW1=X(CBLO+3)                                                     WIL05840
      IF(KDEX.GE.2.AND.HW1.EQ.-22) GOTO 2011                            WIL05850
      IF(OS(2).NE.0) GOTO 2013                                          WIL05860
      CREATE=OS(1)                                                      WIL05870
      IF(INTDEC(1).EQ.1) CREATE=FOS(1)                                  WIL05880
      GOTO 2012                                                         WIL05890
 2013 SEED(1)=SEED(1)*65539                                             WIL05900
      RN=0.5+SEED(1)*0.2328306E-9                                       WIL05910
      BASE=OS(1)-OS(2)                                                  WIL05920
      IF(BASE.LT.0) CALL ERROR(505,0,&9999)                             WIL05930
      CREATE=BASE+(2*OS(2)*RN+0.5)                                      WIL05940
      GO TO 2012                                                        WIL05950
 2011 BASE=OS(1)                                                        WIL05960
      CREATE=BASE*OS(2)                                                 WIL05970
      IF(INTDEC(2).EQ.1) CREATE=BASE*FOS(2)                             WIL05980
 2012 X(CTRA+3)=CLOCK+CREATE                                            WIL05990
      IF(TRBLO) WRITE(OUT6,9021) CREATE                                 WIL06000
      IF(CREATE.NE.0) GOTO 3104                                         WIL06020
      NOUPD=1                                                           WIL06030
      X(CTRA+2)=X(CTRA+1)                                               WIL06040
      GOTO 3001                                                         WIL06050
 2014 NOUPD=0                                                           WIL06060
      IF(ZRTN.EQ.2314) GOTO 2314                                        WIL06070
      IF(ZRTN.EQ.30) GOTO 30                                            WIL06080
      CALL ERROR(603,7,&9999)                                           WIL06090
C                                          ADVANCE                      WIL06100
 202  FW1=X(CTRA+5)                                                     WIL06110
      CBLO=X(BLOCK+HW1)                                                 WIL06120
      FW1=X(CBLO+3)                                                     WIL06130
```

```
      IF(HW1.EQ.-22) GOTO 2021                                          WIL06140
      SEED(1)=SEED(1)*65539                                             WIL06150
      RN=0.5+SEED(1)*0.2328306E-9                                       WIL06160
      BASE=OS(1)-OS(2)                                                  WIL06170
      IF(BASE.LT.0) CALL ERROR(505,0,&9999)                             WIL06180
      DELAY=BASE+(2*OS(2)*RN+0.5)                                       WIL06190
      GOTO 2022                                                         WIL06200
 2021 BASE=OS(1)                                                        WIL06210
      IF(INTDEC(1).EQ.1) BASE=FOS(1)                                    WIL06220
      IF(INTDEC(2).EQ.1) DELAY=BASE*FOS(2)                              WIL06230
 2022 X(CTRA+3)=CLOCK+DELAY                                             WIL06240
      GOTO 230                                                          WIL06250
C                                                   TRANSFER            WIL06260
 203  NEXBLO=OS(1)                                                      WIL06270
      IF(NEXBLO.LT.1.OR.NEXBLO.GT.NBLC) CALL ERROR(603,8,&9999)         WIL06280
      GOTO 231                                                          WIL06290
C                                                   ASSIGN              WIL06300
 204  TPAR=OS(1)                                                        WIL06310
      TASGN=OS(2)                                                       WIL06320
      IF(TPAR.GT.NPAR) CALL ERROR(492,0,&9999)                          WIL06330
      IF(KDEX.LE.2) GOTO 2042                                           WIL06340
      TASGN=TASGN*OS(3)                                                 WIL06350
      IF(INTDEC(3).EQ.1) TASGN=TASGN*FOS(3)                             WIL06360
 2042 IF(MDFR-2) 2043,2044,2045                                         WIL06370
 2043 X(CTRA+8+TPAR)=X(CTRA+8+TPAR)-TASGN                               WIL06380
      GOTO 2046                                                         WIL06390
 2044 X(CTRA+8+TPAR)=TASGN                                              WIL06400
      GOTO 2046                                                         WIL06410
 2045 X(CTRA+8+TPAR)=X(CTRA+8+TPAR)+TASGN                               WIL06420
 2046 GOTO 230                                                          WIL06430
C                                                   MARK                WIL06440
 205  IF(OS(1).GT.NPAR) CALL ERROR(492,0,&9999)                         WIL06450
      IF(OS(1).EQ.0) GOTO 2051                                          WIL06460
      X(CTRA+8+OS(1))=ACLOCK                                            WIL06470
      GOTO 230                                                          WIL06480
 2051 X(CTRA+4)=ACLOCK                                                  WIL06490
      GOTO 230                                                          WIL06500
C                                                   ENTER               WIL06510
 206  IF(OS(1).GT.NSTO) CALL ERROR(499,0,&9999)                         WIL06520
      CSTO=X(STO+OS(1))                                                 WIL06530
      AVAIL=X(CSTO+2)                                                   WIL06540
      WANT=OS(2)                                                        WIL06550
      IF(WANT.EQ.0) WANT=1                                              WIL06560
      IF(WANT.GT.AVAIL) GOTO 2064                                       WIL06570
      USED=X(CSTO+1)                                                    WIL06580
      USED=USED+WANT                                                    WIL06590
      AVAIL=AVAIL-WANT                                                  WIL06600
      MAXCTS=X(CSTO+7)                                                  WIL06610
```

```
      IF(USED.GT.MAXCTS) X(CSTO+7)=USED                                 WIL06620
      X(CSTO+1)=USED                                                    WIL06630
      X(CSTO+2)=AVAIL                                                   WIL06640
      X(CSTO+6)=X(CSTO+6)+WANT                                          WIL06650
      DTIME=CLOCK-X(CSTO+5)                                             WIL06660
      X(CSTO+5)=CLOCK                                                   WIL06670
      FW1=X(CSTO+3)                                                     WIL06680
      FW2=X(CSTO+4)                                                     WIL06690
      DW1=DW1+DTIME*(USED-WANT)                                         WIL06700
      X(CSTO+3)=FW1                                                     WIL06710
      X(CSTO+4)=FW2                                                     WIL06720
      FW1=X(CSTO+8)                                                     WIL06730
      SF=HW1                                                            WIL06740
      SNE=HW2                                                           WIL06750
      IF(SF.EQ.0.OR.AVAIL.NE.0) GOTO 2063                               WIL06760
      TEST=SF                                                           WIL06770
 2061 IF(TEST.EQ.0) GOTO 2062                                           WIL06780
      FW1=X(TEST+6)                                                     WIL06790
      BYTE(2)=.TRUE.                                                    WIL06800
      X(TEST+6)=FW1                                                     WIL06810
      FW1=X(TEST+1)                                                     WIL06820
      TEST=HW1                                                          WIL06830
      GOTO 2061                                                         WIL06840
 2062 SCFLAG=1                                                          WIL06850
 2063 IF(SNE.EQ.0.OR.USED.EQ.0) GOTO 230                                WIL06860
      TEST=SNE                                                          WIL06870
      SNE=0                                                             WIL06880
      GOTO 2061                                                         WIL06890
 2064 FW1=X(CTRA+1)                                                     WIL06900
      FW2=X(CSTO+10)                                                    WIL06910
      HW1=HW3                                                           WIL06920
      HW3=CTRA                                                          WIL06930
      X(CTRA+1)=FW1                                                     WIL06940
      X(CSTO+10)=FW2                                                    WIL06950
      FW1=X(CTRA+6)                                                     WIL06960
      BYTE(2)=.FALSE.                                                   WIL06970
      X(CTRA+6)=FW1                                                     WIL06980
      GOTO 4002                                                         WIL06990
C                                                LEAVE                  WIL07000
 207  IF(OS(1).GT.NSTO) CALL ERROR(499,0,&9999)                         WIL07010
      CSTO=X(STO+OS(1))                                                 WIL07020
      USED=X(CSTO+1)                                                    WIL07030
      GIVEUP=OS(2)                                                      WIL07040
      IF(GIVEUP.EQ.0) GIVEUP=1                                          WIL07050
      IF(GIVEUP.GT.USED) CALL ERROR(425,0,&9999)                        WIL07060
      AVAIL=X(CSTO+2)                                                   WIL07070
      USED=USED-GIVEUP                                                  WIL07080
      AVAIL=AVAIL+GIVEUP                                                WIL07090
```

```
      X(CSTO+1)=USED                                                    WIL07100
      X(CSTO+2)=AVAIL                                                   WIL07110
      DTIME=CLOCK-X(CSTO+5)                                             WIL07120
      X(CSTO+5)=CLOCK                                                   WIL07130
      FW1=X(CSTO+3)                                                     WIL07140
      FW2=X(CSTO+4)                                                     WIL07150
      DW1=DW1+DTIME*(USED+GIVEUP)                                       WIL07160
      X(CSTO+3)=FW1                                                     WIL07170
      X(CSTO+4)=FW2                                                     WIL07180
      FW1=X(CSTO+9)                                                     WIL07190
      SE=HW1                                                            WIL07200
      SNF=HW2                                                           WIL07210
      FW1=X(CSTO+10)                                                    WIL07220
      SRED=HW1                                                          WIL07230
      IF(SE.EQ.0.OR.USED.GT.0) GOTO 2073                                WIL07240
      TEST=SE                                                           WIL07250
 2071 IF(TEST.EQ.0) GOTO 2072                                           WIL07260
      FW1=X(TEST+6)                                                     WIL07270
      BYTE(2)=.TRUE.                                                    WIL07280
      X(TEST+6)=FW1                                                     WIL07290
      FW1=X(TEST+1)                                                     WIL07300
      TEST=HW1                                                          WIL07310
      GOTO 2071                                                         WIL07320
 2072 SCFLAG=1                                                          WIL07330
 2073 IF(SNF.EQ.0.OR.AVAIL.EQ.0) GOTO 2074                              WIL07340
      TEST=SNF                                                          WIL07350
      SNF=0                                                             WIL07360
      GOTO 2071                                                         WIL07370
 2074 IF(SRED.EQ.0.OR.GIVEUP.EQ.0) GOTO 230                             WIL07380
      TEST=SRED                                                         WIL07390
      SRED=0                                                            WIL07400
      GOTO 2071                                                         WIL07410
C                                                   QUEUE               WIL07420
 208  IF(OS(1).GT.NQUE) CALL ERROR(500,0,&9999)                         WIL07430
      CQUE=X(QUE+OS(1))                                                 WIL07440
      UNITS=OS(2)                                                       WIL07450
      IF(UNITS.EQ.0) UNITS=1                                            WIL07460
      PCONTS=X(CQUE+6)                                                  WIL07470
      X(CQUE+2)=X(CQUE+2)+UNITS                                         WIL07480
      X(CQUE+6)=PCONTS+UNITS                                            WIL07490
      MAXCTS=X(CQUE+7)                                                  WIL07500
      IF(X(CQUE+6).GT.MAXCTS) X(CQUE+7)=X(CQUE+6)                       WIL07510
      DTIME=CLOCK-X(CQUE+1)                                             WIL07520
      X(CQUE+1)=CLOCK                                                   WIL07530
      FW1=X(CQUE+4)                                                     WIL07540
      FW2=X(CQUE+5)                                                     WIL07550
      DW1=DW1+DTIME*PCONTS                                              WIL07560
      X(CQUE+4)=FW1                                                     WIL07570
```

```
      X(CQUE+5)=FW2                                                     WIL07580
      FW1=X(CTRA+6)                                                     WIL07590
      HW2=OS(1)                                                         WIL07600
      X(CTRA+6)=FW1                                                     WIL07610
      X(CTRA+7)=CLOCK                                                   WIL07620
      GOTO 230                                                          WIL07630
C                                          DEPART                       WIL07640
  209 IF(OS(1).GT.NQUE) CALL ERROR(500.0.&9999)                         WIL07650
      CQUE=X(QUE+OS(1))                                                 WIL07660
      UNITS=OS(2)                                                       WIL07670
      IF(UNITS.EQ.0) UNITS=1                                            WIL07680
      PCONTS=X(CQUE+6)                                                  WIL07690
      IF(UNITS.GT.X(CQUE+6)) CALL ERROR(428.0.&9999)                    WIL07700
      IF(X(CTRA+7).EQ.CLOCK) X(CQUE+3)=X(CQUE+3)+UNITS                  WIL07710
      X(CQUE+6)=X(CQUE+6)-UNITS                                         WIL07720
      DTIME=CLOCK-X(CQUE+1)                                             WIL07730
      FW1=X(CQUE+4)                                                     WIL07740
      FW2=X(CQUE+5)                                                     WIL07750
      DW1=DW1+DTIME*PCONTS                                              WIL07760
      X(CQUE+4)=FW1                                                     WIL07770
      X(CQUE+5)=FW2                                                     WIL07780
      GOTO 230                                                          WIL07790
C                                          TERMINATE                    WIL07800
  210 IF(.NOT.TRBLO) GOTO 2105                                          WIL07810
      WRITE(OUTX,9041) TRMCNT                                           WIL07820
      KK=8+NPAR                                                         WIL07830
      DO 2106 I=1,KK                                                    WIL07840
      FW1=X(CTRA+I)                                                     WIL07850
      WRITE(OUTX,9019) I.FW1.HW1.HW2                                    WIL07860
 2106 CONTINUE                                                          WIL07870
 2105 FW1=X(CTRA+2)                                                     WIL07880
      IF(HW1.EQ.0) GOTO 2102                                            WIL07890
      FW2=X(HW1+2)                                                      WIL07900
      HW4=HW2                                                           WIL07910
      X(HW1+2)=FW2                                                      WIL07920
 2101 IF(HW2.EQ.0) GOTO 2103                                            WIL07930
      FW2=X(HW2+2)                                                      WIL07940
      HW3=HW1                                                           WIL07950
      X(HW2+2)=FW2                                                      WIL07960
      GOTO 2104                                                         WIL07970
 2102 FTCEC=HW2                                                         WIL07980
      GOTO 2101                                                         WIL07990
 2103 LTCEC=HW1                                                         WIL08000
 2104 HW4=FTRA                                                          WIL08010
      X(CTRA+1)=FW2                                                     WIL08020
      FTRA=CTRA                                                         WIL08030
      TRMCNT=TRMCNT-OS(1)                                               WIL08040
      IF(TRMCNT.LE.0) GO TO 9999                                        WIL08050
```

```
      ZRTN=4005                                                         WIL08060
      GOTO 2311                                                         WIL08070
C                                                   TEST                WIL08080
211   GOTO (2111,2112,2113,2114,2115,2116),MDFR                         WIL08090
C                                                   TEST EQ             WIL08100
2111  IF(OS(1).EQ.OS(2)) GOTO 230                                       WIL08110
      GOTO 2117                                                         WIL08120
C                                                   TEST LT             WIL08130
2112  IF(OS(1).LT.OS(2)) GOTO 230                                       WIL08140
      GOTO 2117                                                         WIL08150
C                                                   TEST LE             WIL08160
2113  IF(OS(1).LE.OS(2)) GOTO 230                                       WIL08170
      GOTO 2117                                                         WIL08180
C                                                   TEST GT             WIL08190
2114  IF(OS(1).GT.OS(2)) GOTO 230                                       WIL08200
      GOTO 2117                                                         WIL08210
C                                                   TEST GE             WIL08220
2115  IF(OS(1).GE.OS(2)) GOTO 230                                       WIL08230
      GOTO 2117                                                         WIL08240
C                                                   TEST U              WIL08250
2116  IF(OS(1).NE.OS(2)) GOTO 230                                       WIL08260
2117  ALTBLO=OS(3)                                                      WIL08270
      IF(ALTBLO.EQ.0) GOTO 4002                                         WIL08280
      NEXBLO=ALTBLO                                                     WIL08290
      GOTO 231                                                          WIL08300
C                                                   TABULATE            WIL08310
212   IF(OS(2).GT.NTAB) CALL ERROR(435,0,&9999)                         WIL08320
      RARG=OS(1)                                                        WIL08330
      IF(INTDEC(1).EQ.1) RARG=FOS(1)                                    WIL08340
      TBLNO=OS(2)                                                       WIL08350
      WTFAC=OS(3)                                                       WIL08360
      IF(INTDEC(3).EQ.1) WTFAC=FOS(3)                                   WIL08370
      IF(WTFAC.EQ.0) WTFAC=1                                            WIL08380
      N=X(TAB+TBLNO)                                                    WIL08390
      RX(N+1)=RX(N+1)+RARG                                              WIL08400
      RX(N+2)=RX(N+2)+(RARG**2)                                         WIL08410
      RX(N+3)=RX(N+3)+(RARG*WTFAC)                                      WIL08420
      RX(N+4)=RX(N+4)+(RARG**2)*WTFAC                                   WIL08430
      X(N+5)=X(N+5)+WTFAC                                               WIL08440
      FREWDT=X(N+6)                                                     WIL08450
      MRNG=X(N+8)-1                                                     WIL08460
      ULLI=X(N+9)                                                       WIL08470
      PNTF=N+9                                                          WIL08480
      DO 2121 I=1,MRNG                                                  WIL08490
      J=I                                                               WIL08500
      IF(RARG.LE.ULLI) GOTO 2122                                        WIL08510
      ULLI=ULLI+FREWDT                                                  WIL08520
2121  CONTINUE                                                          WIL08530
```

```
      RX(N+7)=RX(N+7)+RARG                                              WIL08540
      J=J+1                                                             WIL08550
 2122 X(PNTF+J)=X(PNTF+J)+WTFAC                                         WIL08560
      GOTO 230                                                          WIL08570
C                                                   SAVEVALUE           WIL08580
 213  IF(OS(1).GT.NSAV) CALL ERROR(433.0,&9999)                         WIL08590
      IF(MDFR-2) 2131,2132,2133                                         WIL08600
 2131 X(SAVE+OS(1))=X(SAVE+OS(1))-CS(2)                                 WIL08610
      GOTO 230                                                          WIL08620
 2132 X(SAVE+OS(1))=OS(2)                                               WIL08630
      GOTO 230                                                          WIL08640
 2133 X(SAVE+OS(1))=X(SAVE+OS(1))+OS(2)                                 WIL08650
      GOTO 230                                                          WIL08660
 214  CALL ERROR(603,9,&9999)                                           WIL08670
 215  CALL ERROR(603,10,&9999)                                          WIL08680
C                                                   GATE                WIL08690
 216  GOTO (217,218,219,220),MDFR                                       WIL08700
C                                                   GATE SNE            WIL08710
 217  CSTO=X(STO+OS(1))                                                 WIL08720
      ALTBLO=OS(2)                                                      WIL08730
      USED=X(CSTO+1)                                                    WIL08740
      IF(USED.NE.0) GOTO 230                                            WIL08750
      IF(ALTBLO.EQ.0) GOTO 2171                                         WIL08760
      NEXBLO=ALTBLO                                                     WIL08770
      GOTO 231                                                          WIL08780
 2171 FW1=X(CSTO+8)                                                     WIL08790
      FW2=X(CTRA+1)                                                     WIL08800
      HW3=HW2                                                           WIL08810
      HW2=CTRA                                                          WIL08820
 2172 X(CSTO+8)=FW1                                                     WIL08830
      X(CTRA+1)=FW2                                                     WIL08840
      FW1=X(CTRA+6)                                                     WIL08850
      BYTE(2)=.FALSE.                                                   WIL08860
      X(CTRA+6)=FW1                                                     WIL08870
      GOTO 4002                                                         WIL08880
C                                                   GATE SF             WIL08890
 218  CSTO=X(STO+OS(1))                                                 WIL08900
      ALTBLO=OS(2)                                                      WIL08910
      AVAIL=X(CSTO+2)                                                   WIL08920
      IF(AVAIL.EQ.0) GOTO 230                                           WIL08930
      IF(ALTBLO.EQ.0) GOTO 2181                                         WIL08940
      NEXBLO=ALTBLO                                                     WIL08950
      GOTO 231                                                          WIL08960
 2181 FW1=X(CSTO+8)                                                     WIL08970
      FW2=X(CTRA+1)                                                     WIL08980
      HW3=HW1                                                           WIL08990
      HW1=CTRA                                                          WIL09000
      GOTO 2172                                                         WIL09010
```

```
C                                                GATE SNF
 219  CSTO=X(STO+OS(1))
      ALTBLO=OS(2)
      AVAIL=X(CSTO+2)
      IF(AVAIL.NE.0) GOTO 230
      IF(ALTBLC.EC.0) GOTO 2191
      NEXBLO=ALTBLO
      GOTO 231
 2191 FW1=X(CSTO+9)
      FW2=X(CTRA+1)
      HW3=HW2
      HW2=CTRA
      GOTO 2172
C                                                GATE SE
 22C  CSTO=X(STO+OS(1))
      ALTBLO=OS(2)
      USED=X(CSTO+1)
      IF(USED.EQ.0) GOTO 230
      IF(ALTBLO.EQ.0) GOTO 2201
      NEXBLO=ALTBLO
      GOTO 231
 2201 FW1=X(CSTO+9)
      FW2=X(CTRA+1)
      HW3=HW1
      HW1=CTRA
      GCTO 2172
C                                                SELECT
 221  TPAR=OS(1)
      LOW=OS(2)
      HIGH=OS(3)
      CCMVAL=OS(4)
      ALTBLO=OS(6)
      IF(INTDEC(4).EQ.1) COMVAL=FOS(4)
      TRTN=ZRTN
      ZRTN=2211
      TEMP1=X(CBLO+1)
      TEMP2=X(CBLC+2)
      DO 2230 I=LOW,HIGH
      HW1=-33
      HW2=0
      X(CBLO+2)=FW1
      FW1=X(CBLO+6)
      HW2=I
      X(CBLO+1)=FW1
      INST=CBLO+1
      GOTO 100
 2211 J=0
      GOTO (2213,2214,2215,2216,2217,2218,2219,2220,
```

```
     -2221,2222,2223,2224),MDFR                                        WIL09500
2213 IF(COMVAL.EQ.OS(1)) GOTO 2231                                     WIL09510
     GOTO 2230                                                         WIL09520
2214 IF(COMVAL.LT.OS(1)) GOTO 2231                                     WIL09530
     GOTO 2230                                                         WIL09540
2215 IF(COMVAL.LE.OS(1)) GOTO 2231                                     WIL09550
     GOTO 2230                                                         WIL09560
2216 IF(COMVAL.GT.OS(1)) GOTO 2231                                     WIL09570
     GOTO 2230                                                         WIL09580
2217 IF(COMVAL.GE.OS(1)) GOTO 2231                                     WIL09590
     GOTO 2230                                                         WIL09600
2218 IF(COMVAL.NE.OS(1)) GOTO 2231                                     WIL09610
     GOTO 2230                                                         WIL09620
2219 IF(I.EQ.HIGH) GOTO 2231                                           WIL09630
     IF(J.GE.OS(1)) GOTO 2231                                          WIL09640
     J=OS(1)                                                           WIL09650
     K=I                                                               WIL09660
     GOTO 2230                                                         WIL09670
2220 IF(I.EQ.HIGH) GOTO 2231                                           WIL09680
     IF(J.LE.OS(1)) GOTO 2231                                          WIL09690
     J=OS(1)                                                           WIL09700
     K=I                                                               WIL09710
     GOTO 2230                                                         WIL09720
2221 IF(OS(1).GT.0) GOTO 2231                                          WIL09730
     GOTO 2230                                                         WIL09740
2222 IF(OS(1).EQ.0) GOTO 2231                                          WIL09750
     GOTO 2230                                                         WIL09760
2223 IF(OS(1).NE.0) GOTO 2231                                          WIL09770
     GOTO 2230                                                         WIL09780
2224 IF(OS(1).EQ.0) GOTO 2231                                          WIL09790
     GOTO 2230                                                         WIL09800
2230 CONTINUE                                                          WIL09810
2231 X(CBLO+1)=TEMP1                                                   WIL09820
     X(CBLO+2)=TEMP2                                                   WIL09830
     X(CTRA+8+TPAR)=J                                                  WIL09840
     IF(ALTBLO.EQ.0) GOTO 230                                          WIL09850
     NEXBLO=ALTBLO                                                     WIL09860
     GOTO 231                                                          WIL09870
229  CALL ERROR(603,11,&9999)                                          WIL09880
C                                RETURN FROM BLOCK ROUTINES            WIL09890
230  FW1=X(CTRA+5)                                                     WIL09900
     FW2=X(CTRA+1)                                                     WIL09910
     JBL=HW4                                                           WIL09920
     HW4=HW1                                                           WIL09930
     HW1=HW1+1                                                         WIL09940
     GOTO 2311                                                         WIL09950
231  FW1=X(CTRA+5)                                                     WIL09960
     FW2=X(CTRA+1)                                                     WIL09970
```

```
      JBL=HW4                                                           WIL09980
      HW4=HW1                                                           WIL09990
      HW1=NEXBLO                                                        WIL10000
 2311 X(CTRA+5)=FW1                                                     WIL10010
      X(CTRA+1)=FW2                                                     WIL10020
      N=X(BLOCK+JBL)                                                    WIL10030
      FW1=X(N+1)                                                        WIL10040
      IF(TRBLO) WRITE(OUT6,9040) HW1                                    WIL10050
      IF(HW1.EQ.1) GOTO 2313                                            WIL10060
 2312 IF(.NOT.TRBLO) GOTO 2315                                          WIL10070
      WRITE(OUTX,9031)                                                  WIL10080
      WRITE(OUTX,9032) CTRA,FTRA,FTCEC,LTCEC,FTFEC,LTFEC                WIL10090
 2315 IF(ZRTN.EQ.4005) GOTO 4005                                        WIL10100
      CALL ERROR(603,12,&9999)                                          WIL10110
 2313 WRTN=ZRTN                                                         WIL10120
      TTRA=CTRA                                                         WIL10130
      TBLO=CBLO                                                         WIL10140
      ZRTN=2316                                                         WIL10150
      INST=N+2                                                          WIL10160
      GOTO 100                                                          WIL10170
 2316 OPCODE=1                                                          WIL10180
      CBLO=X(BLOCK+JBL)                                                 WIL10190
      ZRTN=2314                                                         WIL10200
      GOTO 200                                                          WIL10210
 2314 CTRA=TTRA                                                         WIL10220
      CBLO=TBLO                                                         WIL10230
      ZRTN=WRTN                                                         WIL10240
      GOTO 2312                                                         WIL10250
C                                          UPDATE CLOCK MOVE TRANS FROM WIL10260
C                                          FEC TO CEC                   WIL10270
 300  IF(FTFEC.EQ.0) CALL ERROR(401,0,&9999)                            WIL10280
      CLOCK=X(FTFEC+3)                                                  WIL10290
      IF(TRACE) WRITE(OUT6,9023) CLOCK                                  WIL10300
 3001 IF(NOUPD.EQ.1) GOTO 3002                                          WIL10310
      CKPNT= 1                                                          WIL10320
      IF(TRS1) WRITE(OUT6,9042)                                         WIL10330
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL10340
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL10350
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL10360
      CTRA=FTFEC                                                        WIL10370
      FW1=X(CTRA+1)                                                     WIL10380
      FTFEC=HW2                                                         WIL10390
      IF(FTFEC.EQ.0) GOTO 3002                                          WIL10400
      FW1=X(FTFEC+1)                                                    WIL10410
      HW1=0                                                             WIL10420
      X(FTFEC+1)=FW1                                                    WIL10430
 3002 X(CTRA+1)=X(CTRA+2)                                               WIL10440
      CKPNT= 2                                                          WIL10450
```

```
      IF(TRS1) WRITE(OUT6,9042)                                         WIL10460
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL10470
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL10480
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL10490
      IF(FTCEC.LE.0) GOTO 3003                                          WIL10500
      FW1=X(CTRA+5)                                                     WIL10510
      NEWPRI=HW2                                                        WIL10520
      HW2=FTCEC                                                         WIL10530
      GOTO 3004                                                         WIL10540
 3003 FTCEC=CTRA                                                        WIL10550
      CKPNT= 3                                                          WIL10560
      IF(TRS1) WRITE(OUT6,9042)                                         WIL10570
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL10580
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL10590
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL10600
      LTCEC=FTCEC                                                       WIL10610
      X(CTRA+2)=0                                                       WIL10620
      GOTO 3005                                                         WIL10630
 3004 PNTT=HW2                                                          WIL10640
      CKPNT= 4                                                          WIL10650
      IF(TRS1) WRITE(OUT6,9042)                                         WIL10660
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL10670
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL10680
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL10690
      FW2=X(PNTT+5)                                                     WIL10700
      OLDPRI=HW4                                                        WIL10710
      FW1=X(PNTT+2)                                                     WIL10720
      IF(HW2.EQ.0) GOTO 3006                                            WIL10730
      IF(NEWPRI.GE.OLDPRI) GOTO 3004                                    WIL10740
      FW2=X(CTRA+2)                                                     WIL10750
      HW4=HW2                                                           WIL10760
      HW3=PNTT                                                          WIL10770
      X(CTRA+2)=FW2                                                     WIL10780
      FW2=X(HW2+2)                                                      WIL10790
      HW3=CTRA                                                          WIL10800
      X(HW2+2)=FW2                                                      WIL10810
      HW2=CTRA                                                          WIL10820
      X(PNTT+2)=FW1                                                     WIL10830
      IF(TRUPD) WRITE(OUTX,9033)                                        WIL10840
      IF(TRUPD) WRITE(OUTX,9032) PNTT,CTRA,FTCEC,LTCEC,FTFEC,LTFEC      WIL10850
 3005 IF(NOUPD.EQ.1) GOTO 2014                                          WIL10860
      IF(FTFEC.LE.0) GOTO 400                                           WIL10870
      CKPNT= 5                                                          WIL10880
      IF(TRS1) WRITE(OUT6,9042)                                         WIL10890
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL10900
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL10910
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL10920
      NEXBDT=X(FTFEC+3)                                                 WIL10930
```

```
      IF(NEXBDT.GT.CLOCK) GOTO 400                                      WIL10940
      GOTO 3001                                                         WIL10950
 3006 IF(NEWPRI.GE.OLDPRI) GOTO 3007                                    WIL10960
      CKPNT= 6                                                          WIL10970
      IF(TRS1) WRITE(OUT6,9042)                                         WIL10980
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL10990
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL11000
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL11010
      FW2=X(HW1+2)                                                      WIL11020
      HW4=CTRA                                                          WIL11030
      X(HW1+2)=FW2                                                      WIL11040
      FW2=X(CTRA+2)                                                     WIL11050
      HW3=HW1                                                           WIL11060
      HW4=PNTT                                                          WIL11070
      X(CTRA+2)=FW2                                                     WIL11080
      HW1=CTRA                                                          WIL11090
      X(PNTT+2)=FW1                                                     WIL11100
      IF(NOUPD.EQ.1) GOTO 2014                                          WIL11110
      GOTO 3001                                                         WIL11120
 3007 FW2=X(CTRA+2)                                                     WIL11130
      CKPNT= 7                                                          WIL11140
      IF(TRS1) WRITE(OUT6,9042)                                         WIL11150
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL11160
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL11170
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL11180
      HW4=0                                                             WIL11190
      HW3=PNTT                                                          WIL11200
      X(CTRA+2)=FW2                                                     WIL11210
      HW2=CTRA                                                          WIL11220
      X(PNTT+2)=FW1                                                     WIL11230
      LTCEC=CTRA                                                        WIL11240
      IF(NOUPD.EQ.1) GOTO 2014                                          WIL11250
      GOTO 3005                                                         WIL11260
C                                       UNLINK FROM CEC AND MERGE INTO  WIL11270
C                                       FEC                             WIL11280
 310  FW1=X(CTRA+2)                                                     WIL11290
      CKPNT= 8                                                          WIL11300
      IF(TRS1) WRITE(OUT6,9042)                                         WIL11310
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL11320
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL11330
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL11340
      IF(HW1.EQ.0) GOTO 3101                                            WIL11350
      FW2=X(HW1+2)                                                      WIL11360
      HW4=HW2                                                           WIL11370
      X(HW1+2)=FW2                                                      WIL11380
      GOTO 3102                                                         WIL11390
 3101 FTCEC=HW2                                                         WIL11400
      CKPNT= 9                                                          WIL11410
```

```
      IF(TRS1) WRITE(OUT6,9042)                                         WIL11420
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL11430
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL11440
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL11450
 3102 IF(HW2.EQ.0) GOTO 3103                                            WIL11460
      CKPNT=10                                                          WIL11470
      IF(TRS1) WRITE(OUT6,9042)                                         WIL11480
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL11490
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL11500
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL11510
      FW2=X(HW2+2)                                                      WIL11520
      HW3=HW1                                                           WIL11530
      X(HW2+2)=FW2                                                      WIL11540
      GOTO 3104                                                         WIL11550
 3103 LTCEC=HW1                                                         WIL11560
      CKPNT=11                                                          WIL11570
      IF(TRS1) WRITE(OUT6,9042)                                         WIL11580
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL11590
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL11600
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL11610
C                                   UNLINKED                            WIL11620
 3104 X(CTRA+2)=X(CTRA+1)                                               WIL11630
      CKPNT=12                                                          WIL11640
      IF(TRS1) WRITE(OUT6,9042)                                         WIL11650
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL11660
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL11670
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL11680
      IF(FTFEC.EQ.0) GOTO 3108                                          WIL11690
      NEWBDT=X(CTRA+3)                                                  WIL11700
      HW2=FTFEC                                                         WIL11710
 3105 PNTT=HW2                                                          WIL11720
      CKPNT=13                                                          WIL11730
      IF(TRS1) WRITE(OUT6,9042)                                         WIL11740
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL11750
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL11760
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL11770
      OLDBDT=X(PNTT+3)                                                  WIL11780
      FW1=X(PNTT+1)                                                     WIL11790
      IF(HW2.LE.0) GOTO 3107                                            WIL11800
      IF(NEWBDT.GE.OLDBDT) GOTO 3105                                    WIL11810
      IF(HW1.EQ.0) FTFEC=CTRA                                           WIL11820
      HW3=HW1                                                           WIL11830
      HW4=PNTT                                                          WIL11840
      X(CTRA+1)=FW2                                                     WIL11850
      IF(HW3.EQ.0) GOTO 3111                                            WIL11860
      FW3=X(HW3+1)                                                      WIL11870
      HW6=CTRA                                                          WIL11880
      X(HW3+1)=FW3                                                      WIL11890
```

```
      GOTO 3112                                                         WIL11900
 3106 HW4=PNTT                                                          WIL11910
      CKPNT=14                                                          WIL11920
      IF(TRS1) WRITE(OUT6,9042)                                         WIL11930
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL11940
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL11950
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL11960
      HW3=HW1                                                           WIL11970
      X(CTRA+1)=FW2                                                     WIL11980
      IF(HW1.EQ.0) GOTO 3111                                            WIL11990
 3110 FW2=X(HW1+1)                                                      WIL12000
      CKPNT=15                                                          WIL12010
      IF(TRS1) WRITE(OUT6,9042)                                         WIL12020
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL12030
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL12040
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL12050
      HW4=CTRA                                                          WIL12060
      X(HW1+1)=FW2                                                      WIL12070
      GOTO 3112                                                         WIL12080
 3111 FTFEC=CTRA                                                        WIL12090
      CKPNT=16                                                          WIL12100
      IF(TRS1) WRITE(OUT6,9042)                                         WIL12110
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL12120
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL12130
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL12140
 3112 HW1=CTRA                                                          WIL12150
      CKPNT=17                                                          WIL12160
      IF(TRS1) WRITE(OUT6,9042)                                         WIL12170
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL12180
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL12190
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL12200
      X(PNTT+1)=FW1                                                     WIL12210
      GOTO 3109                                                         WIL12220
 3107 IF(NEWBDT.LT.OLDBDT) GOTO 3106                                    WIL12230
      CKPNT=18                                                          WIL12240
      IF(TRS1) WRITE(OUT6,9042)                                         WIL12250
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL12260
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL12270
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL12280
      LTFEC=CTRA                                                        WIL12290
      HW4=0                                                             WIL12300
      HW3=PNTT                                                          WIL12310
      X(CTRA+1)=FW2                                                     WIL12320
      HW2=CTRA                                                          WIL12330
      X(PNTT+1)=FW1                                                     WIL12340
      GOTO 3109                                                         WIL12350
 3108 FTFEC=CTRA                                                        WIL12360
      CKPNT=19                                                          WIL12370
```

```
      IF(TRS1) WRITE(OUT6,9042)                                         WIL12380
      IF(TRS1) WRITE(OUT6,9043) CKPNT,FTFEC,LTFEC,FTCEC,LTCEC,          WIL12390
     -CTRA,PNTT,FTRA,X(CTRA+1),X(CTRA+2),X(CTRA+3),X(PNTT+1),X(PNTT+2), WIL12400
     -X(PNTT+3),NEWBDT,OLDBDT                                           WIL12410
      LTFEC=FTFEC                                                       WIL12420
      X(CTRA+1)=0                                                       WIL12430
 3109 IF(ZRTN.EQ.4002) GOTO 4002                                        WIL12440
      IF(ZRTN.EQ.2314) GOTO 2314                                        WIL12450
      IF(ZRTN.EQ.30) GOTO 30                                            WIL12460
      CALL ERROR(603,13,&9999)                                          WIL12470
C                                          SCAN                         WIL12480
 400  IF(TRACE) WRITE(OUT6,9024)                                        WIL12490
      SCFLAG=0                                                          WIL12510
 4001 CTRA=FTCEC                                                        WIL12520
 4008 FW1=X(CTRA+6)                                                     WIL12530
      SSIND=BYTE(2)                                                     WIL12540
      IF(SSIND) GOTO 4003                                               WIL12550
 4002 FW1=X(CTRA+2)                                                     WIL12560
      IF(HW2.EQ.0) GOTO 300                                             WIL12570
      CTRA=HW2                                                          WIL12580
      GOTO 4008                                                         WIL12590
 4003 FW1=X(CTRA+5)                                                     WIL12600
      CBLO=X(BLOCK+HW1)                                                 WIL12610
      INST=CBLO+2                                                       WIL12620
      FW1=X(CBLO+1)                                                     WIL12630
      OPCODE=HW1                                                        WIL12640
      ZRTN=4004                                                         WIL12650
      GOTO 100                                                          WIL12660
 4004 ZRTN=4005                                                         WIL12670
      GOTO 200                                                          WIL12680
 4005 FW1=X(CBLO+1)                                                     WIL12690
      OPCODE=HW1                                                        WIL12700
      IF(TRSCN) WRITE(OUTX,9034)                                        WIL12710
      IF(TRSCN) WRITE(OUTX,9035) SCFLAG,CBLO,OPCODE,INST                WIL12720
      IF(OPCODE.EQ.10) GOTO 4002                                        WIL12730
      IF(SCFLAG.EQ.1) GOTO 400                                          WIL12740
 4006 IF(OPCODE.NE.2) GOTO 4003                                         WIL12750
      NEWBDT=X(CTRA+3)                                                  WIL12760
      IF(NEWBDT.LE.CLOCK) GOTO 4003                                     WIL12770
      IF(TRSCN) WRITE(OUT6,9030) NEWBDT                                 WIL12780
      ZRTN=4002                                                         WIL12790
      GOTO 310                                                          WIL12800
C                                          ERROR CODES AND MESSAGES     WIL12810
C                                           CLEANUP                     WIL12820
 9002 FORMAT(' SLOPE COMPUTATION')                                      WIL12830
 9003 FORMAT(' BEGIN NEW SIMULATION')                                   WIL12840
 9004 FORMAT(' CLEAR-CONTINUE SIMULATION')                              WIL12850
 9005 FORMAT(' RESET-CONTINUE SIMULATION')                              WIL12860
```

134463
Thesis
W6215 Williams
c.2 A GPSS-like simulator
callable from a FORTRAN
program.

134463
Thesis
W6215 Williams
c.2 A GPSS-like simulator
callable from a FORTRAN
program.

DUDLEY KNOX LIBRARY
3 2768 00305734 0